

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Vybrané možnosti Unreal Engineu 4

Possibilities of Unreal Engine 4

Zadání bakalářské práce

Student: **Ondřej Foukal**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Vybrané možnosti Unreal Engineu 4
Possibilities of Unreal Engine 4

Jazyk vypracování: čeština

Zásady pro vypracování:

Moderní herní enginy, díky svým pokročilým možnostem realtime vizualizace, dnes mají široké uplatnění napříč všemi vědními obory a nejsou již využívány pouze pro vývoj počítačových her. Využívají se při realtime vizualizacích, simulacích, nebo je lze použít pro interakci s objekty ve virtuálním prostředí. Hlavním cílem této práce je zaměřit se na vybrané části v unreal engineu 4, které bude nutné nastudovat, ověřit použitelnost a použít v ukázkové aplikaci v rámci práce. Vybrané části konzultujte průběžně s vedoucím práce.

1. Nastudujte vybrané možnosti UE4, zejména se věnujte novějším částem, které dovolují zvýšit realističnost výsledné scény.
2. Zaměřte se, po domluvě s vedoucím práce, na vybrané části tohoto engineu (např. volumetrická mlha, volumetrické osvětlení, vykreslování terénu, materiálové vlastnosti apod.)
3. Ke zvoleným částem vytvořte základní popis a ukázky, které budou demonstrovat jednotlivé výhody. Pokud to bude možné, výsledky porovnejte s jiným řešením.
4. Výsledky této práce by měly navázat na obhájené bakalářské a diplomové práce v minulých letech.
5. Celý postup včetně vývoje aplikace pečlivě zdokumentujte, aby v práci bylo možné dále pokračovat.

Seznam doporučené odborné literatury:

- [1] Carnall, B.: Unreal Engine 4.X By Example. 2016. ISBN 978-1-78588-553-2
- [2] McCaffrey, M.: Unreal Engine VR Cookbook: Developing Virtual Reality with UE4. Addison-Wesley Professional (2017). ISBN 978-01346491.
- [3] Shannon, T.: Unreal Engine 4 for Design Visualization: Developing Stunning Interactive Visualizations, Animations, and Renderings (Game Design). 2017. ISBN 978-0134680705.


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

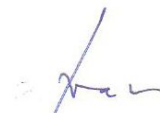
Vedoucí bakalářské práce: **Ing. Martin Němec, Ph.D.**

Datum zadání: 01.09.2018

Datum odevzdání: 30.04.2020




doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry


prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto práci vypracoval samostatně a že jsem uvedl všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 14. května 2020


.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků článku 26, odst.9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 14. května 2020


.....

Na tomto místě bych rád poděkoval Ing. Martinu Němcovi, PhD., za cenné nápady, rady a připomínky, díky nimž mohla tato práce vzniknout. Chtěl bych také poděkovat rodičům a všem svým blízkým, kteří mě ve studiu podporovali.

Abstrakt

Práce se zabývá možnostmi vizualizace budov s využitím moderních herních enginů. První část se zabývá technikami tvorby 3D modelu reálné budovy. Ve druhé části jsou popsány materiály a tvorba textur. Poslední část je věnovaná úpravě modelu v herním enginu, což je například nasvícení scény, vytvoření terénu, vegetace a podobně. Praktické části byly předvedeny na budově fakulty elektrotechniky a informatiky VŠB-TUO.

Klíčová slova: Blender, 3D modelování, Vizualizace, Unreal Engine 4

Abstract

The thesis deals with the possibilities of buildings' visualisation using modern game engines. The first part of the thesis is focused on the creation of a 3D model of an actual building. The second part is dedicated to the description of materials and texture creation. Last but not least, the thesis' final section depicts the adjustment of a model in a game engine, such as scene lighting, terrain or vegetation creation and so on. The practical parts of this thesis were demonstrated on the Faculty of Electrical Engineering and Computer Science VSB-TUO.

Keywords: Blender, 3D modelling, Visualization, Unreal Engine 4

Obsah

| | |
|--|-----------|
| Seznam použitých zkratk a symbolů | 8 |
| Seznam obrázků | 9 |
| 1 Úvod | 11 |
| 2 Tvorba modelu | 12 |
| 2.1 Modelovací aplikace | 12 |
| 2.2 Příprava a sběr materiálů | 14 |
| 2.3 Tvorba modelu budovy fakulty | 17 |
| 2.4 Příprava pro export do FBX | 22 |
| 3 Materiály | 26 |
| 3.1 Specular vs metallic workflow | 26 |
| 3.2 Příprava textur | 27 |
| 3.3 Použité formáty | 29 |
| 4 Herní engine | 31 |
| 4.1 Možnosti osvětlení Unreal Engine 4 | 32 |
| 4.2 Import vytvořeného modelu | 34 |
| 4.3 Nasvícení modelu fakulty | 37 |
| 4.4 Nastavení odrazů a odlesků | 40 |
| 4.5 Pohyb ve scéně | 40 |
| 4.6 Vykreslování terénu a vegetace | 43 |
| 4.7 Volumetrická mlha | 49 |
| 4.8 Interakce se scénou | 50 |
| 4.9 Srovnání podobných prací | 52 |
| 5 Závěr | 54 |
| Literatura | 55 |
| Přílohy | 56 |
| A Seznam přiložených obrázků: | 57 |

Seznam použitých zkratk a symbolů

| | |
|------|---|
| 3D | – Three dimensional |
| BMP | – Bitmap |
| CAD | – Computer Aided Design |
| CC | – Creative Cloud |
| FBX | – FilmBox |
| FEI | – Fakulta Elektrotechniky a Informatiky |
| GB | – Gigabajt |
| JPEG | – Joint Photographic Experts Group |
| JPG | – Joint Photographic Group |
| NPC | – Non-player character |
| PBR | – Physically based rendering |
| PSD | – Photoshop Document |
| RAW | – Z anglického slova syrový |
| RTX | – Raytracing |
| UDIM | – U dimension |
| UE | – Unreal Engine |
| VR | – Virtuální realita |

Seznam obrázků

| | | |
|----|--|----|
| 1 | Subdivision Surface | 13 |
| 2 | Modelovací nástroje Blenderu | 14 |
| 3 | Snímek z Google Maps [10] | 15 |
| 4 | Evakuační plán | 16 |
| 5 | Primitiva v programu blender [6] | 17 |
| 6 | Modelování vnitřních stěn | 19 |
| 7 | Modelování evakuačního schodiště s využitím modifikátoru Array | 20 |
| 8 | Modelování karuselových dveří | 20 |
| 9 | Hotové karuselové dveře | 21 |
| 10 | UV Unwrapping | 22 |
| 11 | Smart UV Project | 23 |
| 12 | Manuální mapování | 23 |
| 13 | Kolizní objekty | 24 |
| 14 | Model před exportem do Unreal Engine 4 | 25 |
| 15 | Nastavení formátu RAW | 28 |
| 16 | Vlevo Base Color mapa, uprostřed Roughness mapa, vpravo Normal mapa | 29 |
| 17 | Vlevo Metallic mapa, uprostřed Alpha mapa, vpravo Emission mapa | 29 |
| 18 | Ukázka volumetrické mlhy z dokumentace Unreal Engine | 33 |
| 19 | Nastavení pro import z FBX | 34 |
| 20 | Scéna po importu z FBX | 35 |
| 21 | Ukázka neprůhledného materiálu linolea | 36 |
| 22 | Ukázka maskovaného materiálu a použití Emission mapy | 36 |
| 23 | Ukázka materiálu průhledného skla | 37 |
| 24 | Doporučený texel density z dokumentace Unreal Engine [21] | 38 |
| 25 | Nastavený texel density pro interiér budovy | 38 |
| 26 | Nastavení lighmass | 38 |
| 27 | Vypočtená globální iluminace bez použití dalších světél | 39 |
| 28 | Vypočtená globální iluminace s přidáním světly uvnitř budovy | 39 |
| 29 | Porovnání Cube Reflection Capture (vlevo) a Sphere Reflection Capture (vpravo) | 41 |
| 30 | Upravený blueprint FirstPersonCharacter | 42 |
| 31 | Ukázka jednoduché a složité kolize z webu UnrealEngine.com | 42 |
| 32 | Ukázka kolizí | 43 |
| 33 | Ukázka NavMeshBoundsVolume a Nav Modifier Volume | 44 |
| 34 | Ukázka Nav Link Proxy | 44 |
| 35 | Nastavení pro tvorbu terénu | 45 |
| 36 | Nastavení štětce | 46 |
| 37 | Upravený terén | 46 |

| | | |
|----|--|----|
| 38 | Vícevrstevný materiál terénu | 47 |
| 39 | Terén s materiálem | 48 |
| 40 | Nastavení pro přidání vegetace | 49 |
| 41 | Výsledný terén | 49 |
| 42 | Test jestli se hráč nachází v oblasti kolizního boxu | 50 |
| 43 | Ukázka dveří a zónou ve které je možné dveře otevřít | 51 |
| 44 | Sekvence otevírání dveří | 51 |
| 45 | Detekce interakce otevření dveří | 51 |
| 46 | Ukázka úkolu | 52 |
| 47 | Pohled ze strany poslucháren | 58 |
| 48 | Pohled ze strany parkoviště | 59 |
| 49 | Pohled ze strany evakuačního schodiště | 60 |
| 50 | Pohled na hlavní vchod | 61 |
| 51 | Pohled na interiér budovy - 2.patro | 62 |

1 Úvod

V dnešní moderní době kolem nás existují počítače, ať už pro osobní použití, renderovací farmy či superpočítače s obrovským výkonem, které se stále vyvíjejí, zdokonalují a zrychlují. Počítače se využívají nejen pro výuku, ale také pro simulace, vizualizace, v herním průmyslu, pro realtime rendering a pro virtuální realitu. Je možné si prohlédnout budovu či instituci na počítači, aniž bychom ji museli reálně navštívit; 3D modely se používají v široké škále oborů. Zdravotnický průmysl používá modely orgánů, filmový průmysl je používá pro výrobu animovaných filmů, trikových záběrů i celých scén ve hraných filmech, modely se používají ve videohrách. Inženýři modelují s pomocí 3D programů nová zařízení. Proto význam vizualizace neustále roste a objevují se nové a nové možnosti jejího použití. Herní engine však samotný dnes nestačí, je potřeba vytvořit modely, textury, materiály, k čemuž se používají další nástroje. Cesta takového modelu prochází několika etapami.

Bakalářská práce se zabývá oblastí architektury. Cílem práce je zaměřit se na možnosti, jak vytvořit model reálné budovy na počítači a následně ho převést do herního enginu a vizualizovat ho tak, aby se návštěvník mohl virtuálně pohybovat po budově. Takovými pracemi se zabýval například student Fakulty informatiky Přírodovědecké fakulty Univerzity Palackého v Olomouci [1], který v roce 2017 vytvořil model tamější Přírodovědecké fakulty. Na jeho práci navázal se svou prací další student, jehož práce pojednávala o interiérech téže fakulty v Unreal Engine 4 [2]. Podobnou práci dále vytvořil student univerzity v estonském Tartu [3], který modeloval budovy v Blenderu a vizualizoval je v enginu Unity. Srovnání méj práce s těmito pracemi uvádím v kapitole 4.9

Práce bude rozčleněna na tři hlavní části. V první z nich se budu zabývat problematikou sběru materiálů, resp. podkladů pro následné praktické vytváření práce, jakož i procesem tvorby vytváření modelu a jeho přípravou pro export. Následně se budu zabývat materiály, tvorbou textur, jakož i použitými grafickými formáty. V návaznosti na to bude pozornost věnována importu výsledného modelu do herního enginu a možnostem vizualizace vytvořené budovy.

Závěr práce bude věnován vyhodnocení výsledků, ke kterým jsem dospěl v rámci praktické části práce a jejího vytváření. Pokusím se rovněž o naznačení dalšího směru, kterým by se tato práce mohla v budoucnu vyvíjet, respektive o způsoby jejího dalšího využití, jakož i případného rozšíření například v rámci diplomové či jiné akademické práce.

2 Tvorba modelu

Před samotným modelováním nastává otázka, jaký program vybrat pro 3D modelování. Rychle se rozšiřuje i počet aplikací, které cílí i na uživatele bez předchozí technické průpravy. K jednodušším patří CAD systémy, například SketchUp či 123D Design, existují i složitější profesionální CAD aplikace, například Fusion 360, OnShape či OpenSCAD. Za funkčně dobře vybavený 3D modelovací nástroj považuji Blender [4], jenž je vyvíjen pod licencí open-source. Díky tomu je volně dostupný všem uživatelům na platformách, jakými jsou Windows, Mac či Linux, a to zdarma. V oblasti placených 3D modelovacích nástrojů patří mezi nejpoužívanější Cinema 4D, Maya, 3Ds Max, Modo, Zbrush, Houdiny.

2.1 Modelovací aplikace

Pro tvorbu modelu fakulty jsem zvolil program Blender. Tento program obsahuje nástroje pro modelování, renderování, animace, simulace, kompozici a sledování pohybu. Ve verzi 2.8 prošel velkými změnami uživatelského rozhraní a stal se tak přívětivějším pro začínající uživatele, protože bylo jeho ovládání přiblíženo konkurenci. Do té doby se totiž co do ovládání v rámci uživatelského rozhraní značně od konkurenčních programů odlišoval, což mnohým začínajícím uživatelům působilo problémy. Mezi nově přidané funkce patří Eevee, což je fyzikálně postavený renderer pracující v reálném čase. Ve verzi 2.82 byla tvůrci přidána i funkce UDIM.

Pro Blender existuje kompletní podrobná dokumentace a velké množství výukových materiálů a videonávodů udržovaných komunitou uživatelů Blender, což z něj spolu s výše zmíněnými změnami dělá, dle mého názoru, skvělou volbu pro začínající i zkušené 3D grafiky [5], [6].

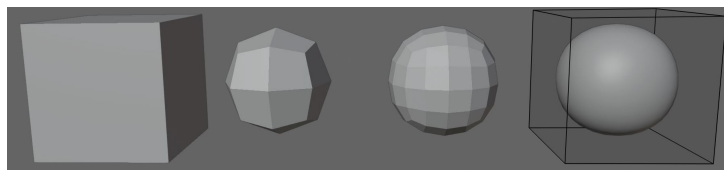
2.1.1 Nástroje a funkce programu Blender

S ohledem na cíl této práce, kterým není popsat práci v programu Blender, zmíním v této pasáži pouze vybrané stěžejní nástroje, které jsem využil při vytváření modelu fakulty. Pro bližší informace o tomto programu a práci v něm viz [5]. V práci jsem se soustředil na tvorbu modelu fakulty, který je výchozím bodem mé práce. V následující pasáži zjednodušeně popíši, jak vybrané části modelu fakulty vznikaly a se kterými problémy jsem se musel při jeho tvorbě vypořádat.

Módy interakce s objektem

Mezi dva nejčastěji používané módy v rámci Blenderu patří Object mode a Edit mode, jakož i Sculpt Mode, Vertex Paint, Weight Paint, Texture Paint. Ve své práci jsem použil tyto tři módy.

- Object mode - V režimu Object mode je možné pouze manipulovat s objektem nástroji Move (posun), Scale (zvětšení) a Rotate (Rotace). V tomto režimu lze také na model aplikovat modifikátory, z nichž jsem však u práce použil jen Subdivision Surface, Array a Solidify.



Obrázek 1: Subdivision Surface

- Edit Mode - umožňuje manipulovat s jednotlivými vertexy, hranami a polygony modelu a používat další nástroje uvedené v kapitole 2.1.1. Používá se pro detailní modelování prakticky jakýchkoliv objektů[7].
- Sculpt mode - Pro tvorbu modelů s velkým počtem polygonů se používá tzv.sculpting. Přirovnává se k sochařství a pomocí nástrojů jako Push, Pull a Smooth se vytváří model v řádu milionu polygonů s velkým množstvím detailů. Ve svém modelu jsem tuto metodu nevyužil.

Další informace najdeme například v dokumentaci Blenderu[8]

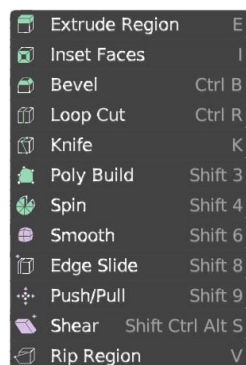
Modifikátory

Modifikátory umožňují na výsledný objekt aplikovat procedurální operace a změnit tak jeho vlastnosti. Na rozdíl od úpravy modelů v editačním režimu je možné modifikátor odebrat a vrátit tak objekt do jeho původního stavu. Modifikátory je také možné na objekt aplikovat, čímž se jejich efekt trvale zapeče do objektu. Po tomto kroku už není možné nastavení modifikátoru upravovat, ani modifikátor odebrat.

- Modifikátor Array - vytváří pole kopií objektu, na který je modifikátor aplikován. Tyto kopie jsou lineárně „posunuty“ ve směrech x, y, z; a to buď relativně dle velikosti objektu, nebo konstantě zadanou hodnotou. Tento modifikátor je vhodný k vytvoření komplexních objektů skládajících se z opakovaných tvarů. V mé práci hrál tento modifikátor významnou roli zejména při vytváření venkovního evakuačního schodiště.
- Subdivision Surface - slouží k rozdělení ploch objektu a jejich zaoblení. Používaný pro tvorbu high-poly objektů. Počet kroků rozdělení objektu je možné nastavit zvlášť pro zobrazení v 3D pohledu a pro render. Na obrázku 1 je ukázka nastavení počtu kroků rozdělení od 0 (vlevo) do 3 (vpravo). Na ukázce vpravo lze také vidět, že při editaci objektu v Edit Mode je objekt stále reprezentovaný jeho původním tvarem.
- Solidify - vytvoří hloubku (tloušťku) u objektu, na který je aplikován.

Nástroje editačního módu

Blender nabízí velké množství nástrojů pro úpravu ploch, hran či vertexů 3D modelů. Ukázky použití těchto nástrojů jsou předvedeny přímo na modelování vybraných částí fakulty v kapitole



Obrázek 2: Modelovací nástroje Blenderu

2.3 Ukázka menu výběru nástrojů na obrázku 2 Mezi nejčastěji použité nástroje v této práci patří.

- Extrude - Základem této metody je postupné vytahování nových hran a ploch objektu v určitém směru. Jedná se o velice častý způsob modelování v Blenderu, kterým lze vytvořit celou řadu objektů[9].
- Inset Face - tento nástroj vloží do oblasti vybraných polygonů zmenšenou verzi tohoto výběru se zachováním původní topologie.
- Bevel - slouží ke tvorbě zaoblených hran a rohů geometrie. Objekty v reálném světě nemají téměř nikdy dokonale ostré hrany.
- Loop Cut - Slouží pro rozdělení polygonů na více segmentů při zachování topologie tělesa.
- Knife - Slouží pro nařezání ploch objektu na menší celky. Na rozdíl od Loop Cut můžeme vést řezy libovolně.

2.2 Příprava a sběr materiálů

Jako první reference pro tvorbu modelu posloužily fotografie exteriéru fakulty nafocené mým mobilním telefonem. Jelikož jsem neměl přístup ke stavebním plánům fakulty, tak jsem na hrubé rozvržení budovy použil snímek z Google map [10] (viz obrázek 4), tedy pohled seshora, podle kterého jsem rozvrhl základní tvar budovy. Přesné rozměry jsem odhadl z měřítka Google map a pak po srovnání s fotografiemi jsem získal výšku budovy.

Po vytvoření základního modelu již bylo třeba budovu natekturovat, aby dostala základní vzhled. K tomu již nestačily fotografie z telefonu, ale byl použit fotoaparát značky Canon.

Tímto fotoaparátem byly nafoceny i obrázky určené pro tvorbu textur. Z toho důvodu bylo při jejich focení podstatné, aby byly foceny za co nejlepšího světla, aby neměly ostré stíny, aby byly fotografované objekty zachyceny co nejdetailněji, jakož i aby byly foceny co nejvíce



Obrázek 3: Snímek z Google Maps [10]

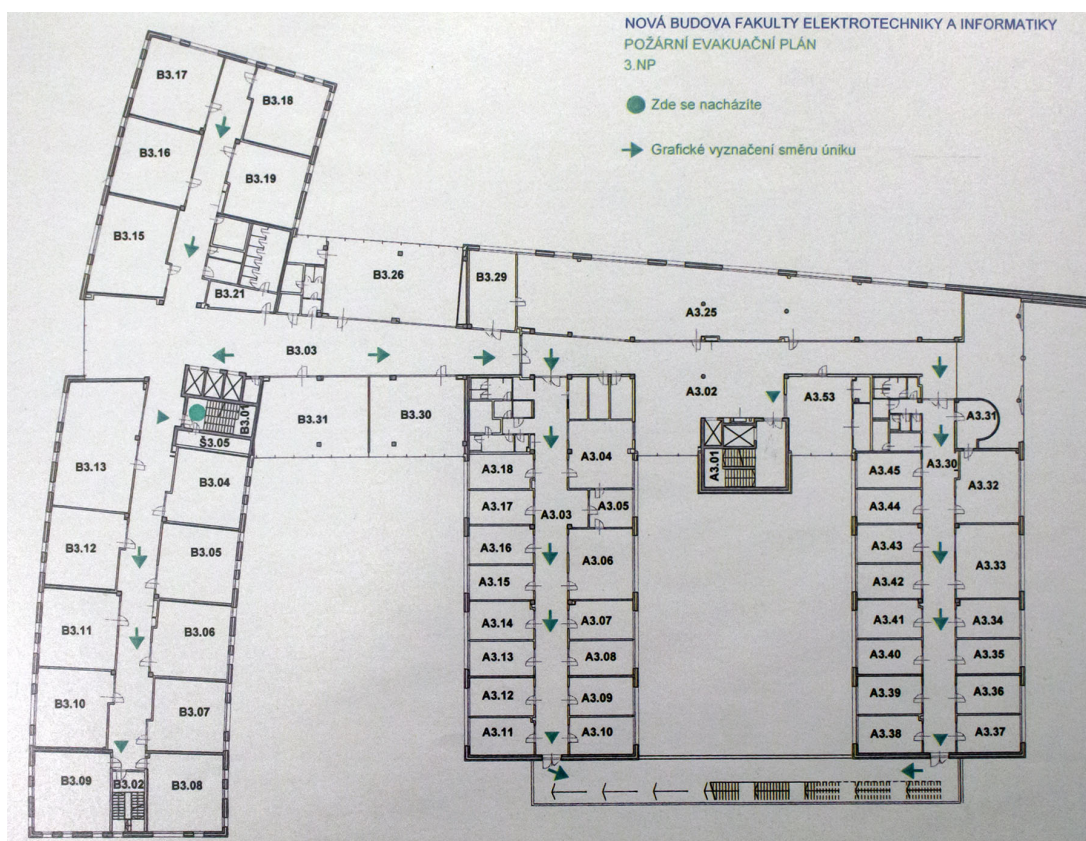
v rovině a z přímého úhlu. Fotografie byly pořizovány ve formátu RAW, jehož výhody jsou popsány v kapitole 3.3. Fotografování proběhlo během několika týdnů, a to ve dvou etapách.

Focení exteriéru

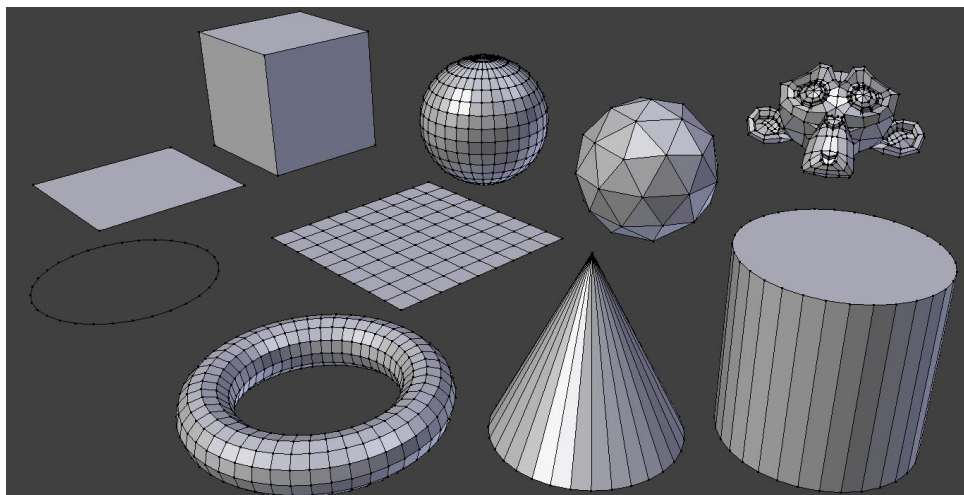
Pro focení venkovní části budovy bylo nejpodstatnější vhodné počasí a správně zvolená denní doba. Jelikož s fotografováním mám minimální zkušenosti, tak jsem zkoušel fotit brzy ráno, odpoledne a před setměním. Nejlépe dopadly fotografie focené po rozbřesku, a proto jsem v tuto dobu následně nafotil i zbytek fotografií. Nastavení fotoaparátu bylo ponecháno na automaticce. Ručně jsem pouze zaostřoval a přibližoval optickým zoomem. Při focení venkovních částí jsem se nesetkal s výraznějšími problémy, fotografie byly většinou ostré a barevně vyvážené s jen minimálním množstvím šumu [11].

Focení interiéru

Při focení interiéru fakulty jsem se setkal s velkým množstvím problémů. Oproti focení exteriérů bylo podstatné zvolit nejen denní dobu, kdy se dostává do budovy dostatek světla, ale taky čas, ve kterém se po budově nepohybuje příliš mnoho studentů, lidí, učitelů atd. Spousta fotografií se ukázala jako nepoužitelná pro tvorbu textur. Fotografie byly neostré, zrnité a se značným množstvím stínů a světel. Nakonec tedy byla většina fotek interiéru použita jen jako referenční obrázky pro tvorbu interiéru.



Obrázek 4: Evakuační plán



Obrázek 5: Primitiva v programu blender [6]

Jen s použitím fotografií by bylo velmi obtížné odhadnout rozměry a rozložení jednotlivých chodeb a místností fakulty. Z toho důvodu jsem nafotil evakuační plán fakulty.

2.3 Tvorba modelu budovy fakulty

Model fakulty se skládá ze stovek samostatných objektů. Ve většině případů se jedná o poměrně jednoduché objekty, avšak jejich velký počet výrazně zvýšil časovou náročnost tvorby praktické části této práce. Jednotlivé objekty byly vytvářeny se zaměřením na kompromis mezi kvalitou a optimalizací, jelikož nebylo možné dopředu odhadnout, jaký bude mít jejich počet a kvalita dopad na výkon výsledné scény po implementaci do Unreal Engine.

2.3.1 Primitiva

Primitiva, která nabízí program Blender, jsou uvedena na obrázku ???. Pro modelování většiny objektů jsem vycházel ze tří typů primitiv. Primitivum plocha sloužilo jako základ pro vnitřní stěny, podlahy, chodníky a silnice. Krychle sloužila pro objekty, jakými jsou rámy dveří, venkovní části budovy, posluchárny, schodiště a podobně. Válec jsem používal pro tvorbu vstupních karuselových dveří, stropních světel, pouličních lamp a sloupů. Model fakulty FEI je tvořen stovkami objektů modelovaných velice podobnými metodami. V této práci uvádím několik zajímavějších příkladů modelování a techniky, které se u nich používaly.

Modelování vnitřních stěn fakulty

Pro tento případ byly použity evakuační plány pro jednotlivá patra jako pozadí pro modelování z pohledu seshora. Nejprve bylo vytvořeno primitivum plocha, které bylo následně zmenšeno tak, aby zabíralo celý prostor evakuačního plánu. Poté byly vytvořeny horizontální a vertikální řezy nástrojem Loop Cut a pomocí nástroje Edge Slide a transformačních nástrojů (Move, Scale,

Rotate) byly hrany a vertexy napasovány tak, aby pasovaly na pozadí. Po tomto kroku byly odstraněny nepotřebné polygony.

Dalším krokem bylo nařezání polygonů nástrojem Knife tak, aby nově vytvořené hrany odpovídaly fotografii na pozadí. Přebytné hrany byly následně odstraněny pomocí funkce Dissolve v režimu pro hrany a polygony. Pro další postup už byl použit pohled z perspektivy. Polygony odpovídající stěnám byly ručně označeny a dvakrát vytaženy do prostoru nástrojem Extrude po směru osy z. První vytažení bylo do výšky dveří a druhé až do výšky stropu. Následně byly nástroji Fill a Bridge vyplněny prostory nad dveřmi a stejným způsobem byl vytvořen strop. Výše popsané kroky jsou znázorněny na obrázku 6.

Modelování evakuačního schodiště

Venkovní evakuační schodiště fakulty se může na první pohled zdát jako relativně komplikovaný objekt. Vhodným použitím modifikátoru Array na jednotlivé části konstrukce však poměrně snadno vznikne komplexně působící výsledný model.

Jednotlivé kovové části konstrukce jsou si velmi podobné. Jako základní primitivum pro ně byla vzhledem k hranatému tvaru použita krychle. Pomocí operací Inset Faces, Loop Cut a Extrude byly upraveny do výsledné podoby.

Celkem je tedy Array modifikátor s různým nastavením použit na 13 unikátních objektech, které tvoří celkový model evakuačního schodiště (viz horní části obrázku 7). Pro horizontální pilíře byl Array modifikátor použit třikrát po sobě pro vytvoření kopií po osách x, y, z. Pro dokončení výsledného objektu bylo ještě třeba aplikovat všechny modifikátory, objekty spojit pomocí funkce Join a ručně odstranit části konstrukce, které protínají schodiště, jak lze vidět ve spodní části obrázku 7.

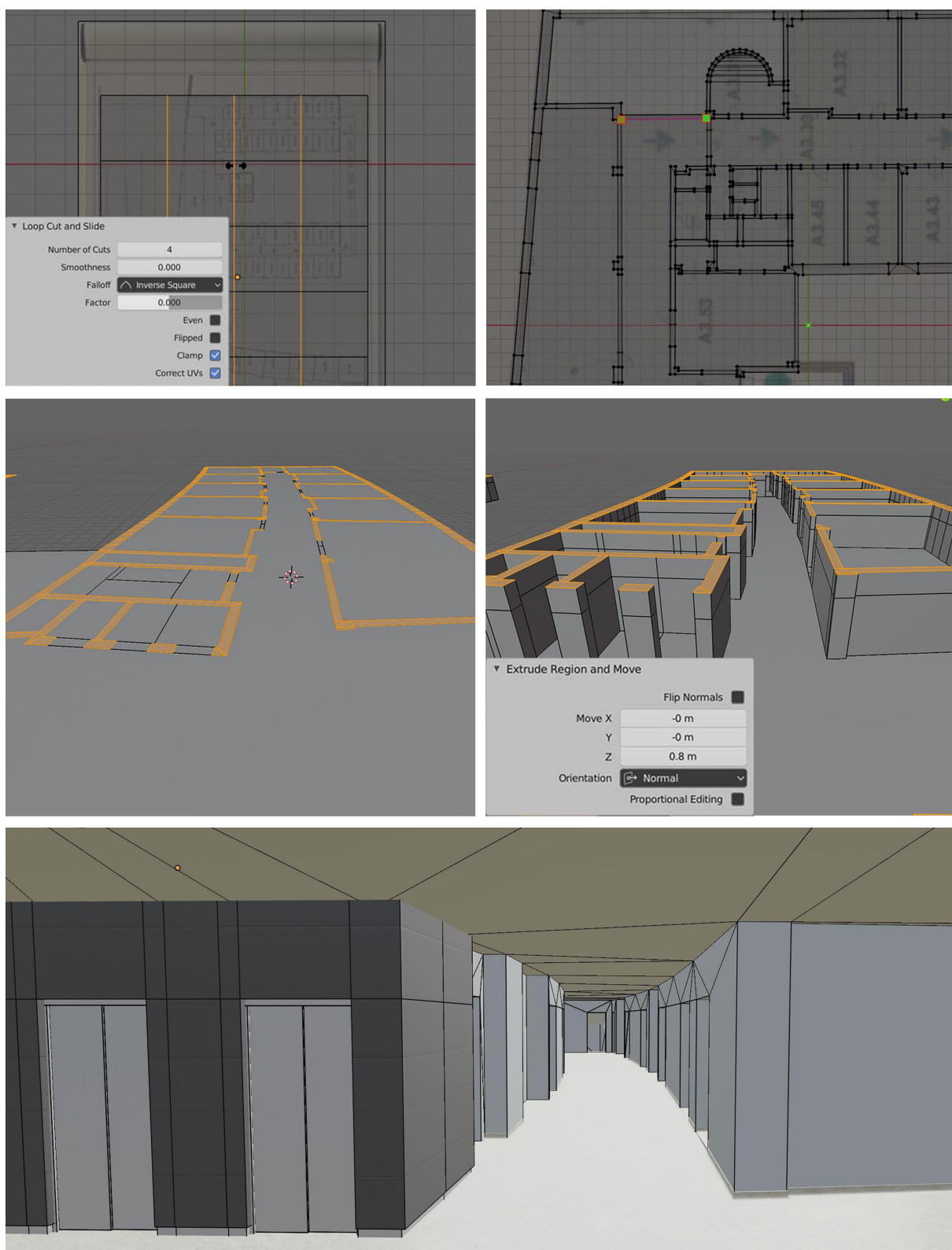
Modelování vstupních karuselových dveří

Pro tvorbu karuselových dveří bylo zvoleno primitivum válec. Při tvorbě grafiky, obzvláště pro renderování v reálném čase, je třeba zvážit, kolik polygonů by měl mít vytvářený model vzhledem k jeho velikosti, důležitosti pro danou scénu a vzdálenosti, z jaké bude objekt viditelný. Jelikož v tomto případě se jedná o velký objekt, kterým se bude procházet, bylo zvoleno 64 vertexů po obvodu.

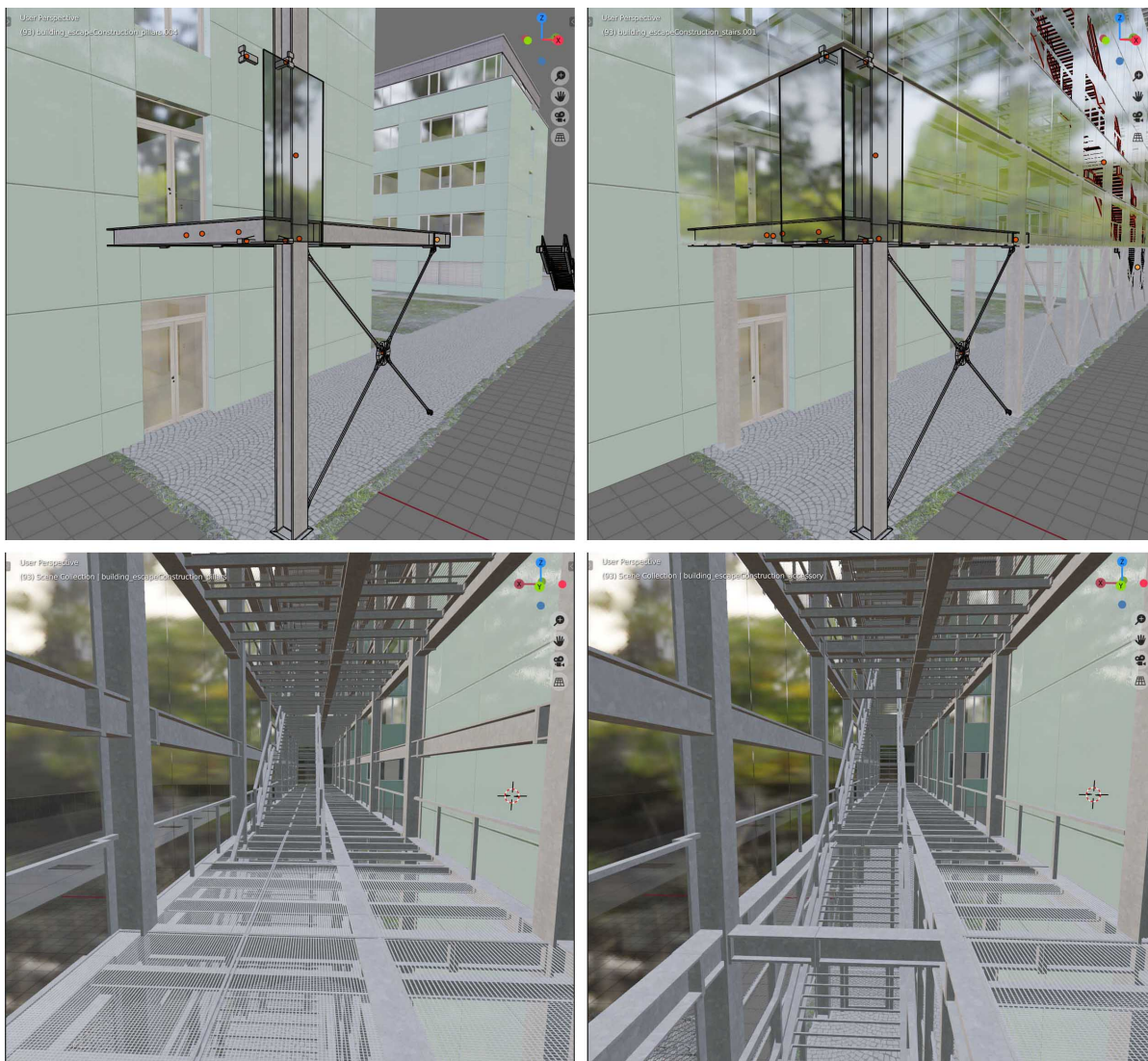
Následně byl nástrojem Loop Cut and Slide rozdělen objekt vertikálně. Polygony pro část, kterou se má procházet, byly odstraněny. Pro vytvoření objemu byl použit modifikátor Solidify (viz obrázek 8 vpravo). Následně byly vytvořeny další detaily pomocí nástrojů Extrude, Inset Faces, Bevel a Knife. Výsledný model karuselových dveří po natexturování je na obrázku 9

2.3.2 UV mapování a texturování

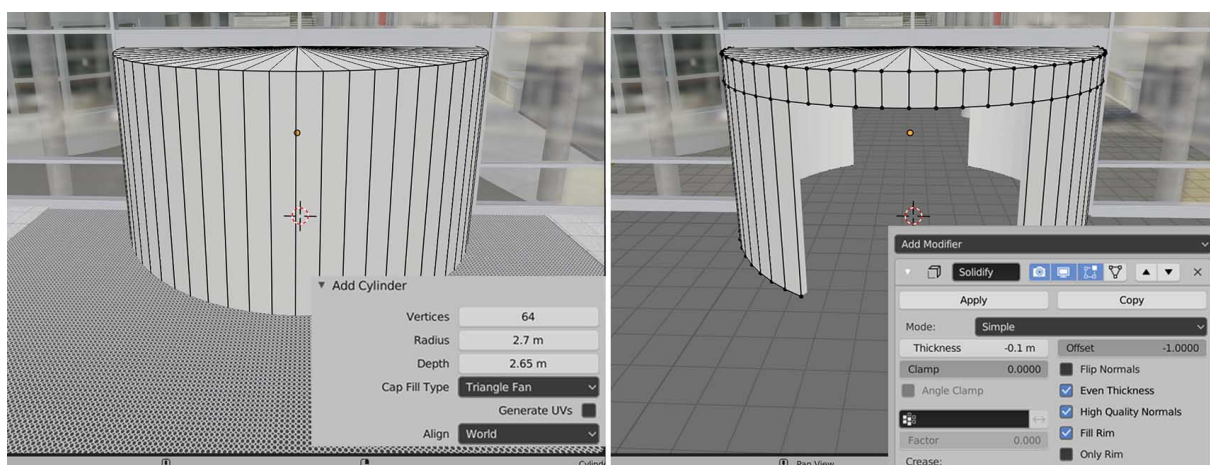
Po vymodelování každého z objektů bylo třeba aplikovat na objekt materiál. Pro tento účel je třeba vytvořit UV mapu objektu (tedy objekt „rozbalit“ do jeho 2D reprezentace. Během této



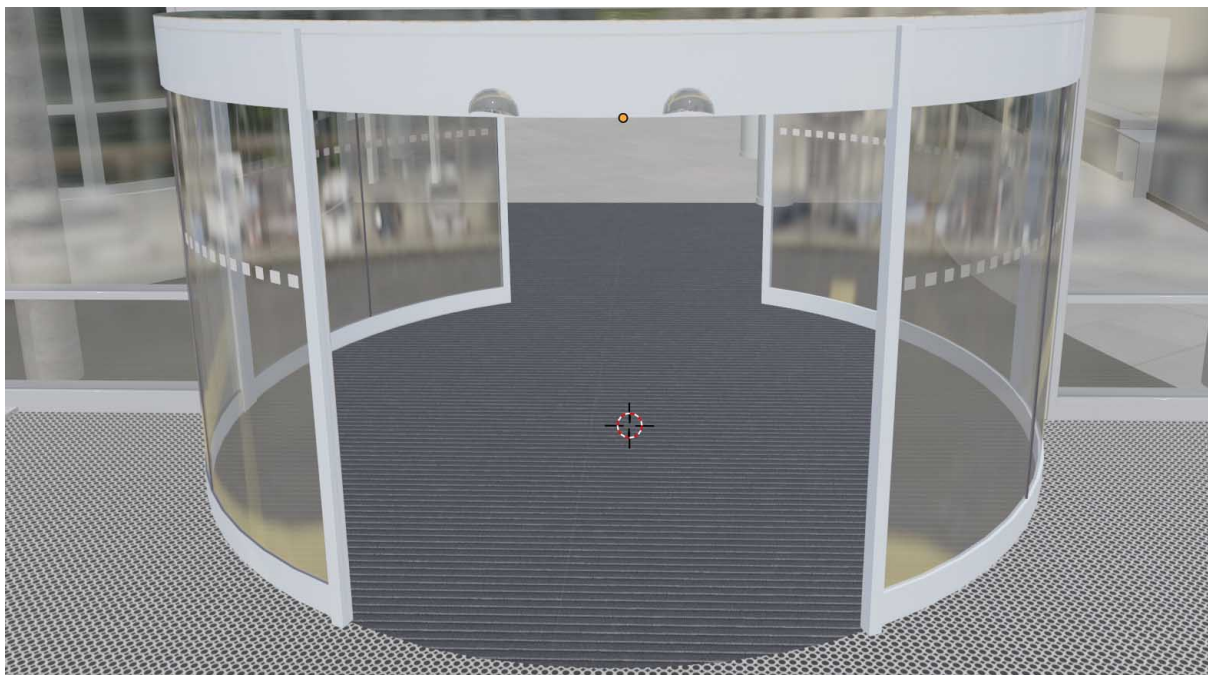
Obrázek 6: Modelování vnitřních stěn



Obrázek 7: Modelování evakuačního schodiště s využitím modifikátoru Array



Obrázek 8: Modelování karuselových dveří



Obrázek 9: Hotové karuselové dveře

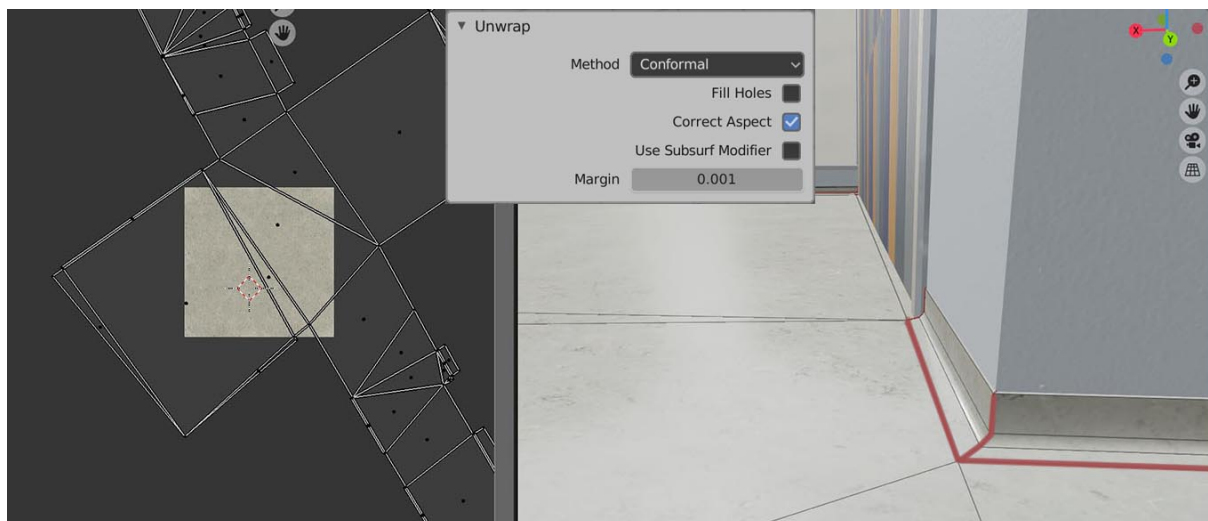
práce byly nejčastěji použity tyto postupy, Smart UV project, UV Unwrapping a „manuální mapování“. Smart UV project a UV unwrapování jsem použil u modelů nebo jejich částí, které používají seamless textury, a dále u všech modelů pro vytvoření druhé UV mapy, do které se bude ukládat lightmap v Unreal Engine. Manuální unwrap byl nejčastěji použit u objektů nebo jejich částí, které používají atlas. V následujícím textu představím na třech ukázkách stručně tyto tři přístupy.

UV Unwrapping

Nástroj UV Unwrap jsem použil pro tvorbu UV mapy vnitřních podlah a dlažby před budovou fakulty. UV Unwrap vyžaduje označit v Edit Mode hrany, podle kterých bude objekt automaticky „rozbalen“ do dvourozměrné reprezentace. Takto označeným hranám se říká Seam. Podle složitosti objektu se jednalo o poměrně časově náročný proces. Na obrázku ?? můžeme vidět červeně označené hrany určené pro řez a výslednou UV mapu zvětšenou za hranice plochy, což znamená, že se bude textura na objektu opakovat.

Smart UV project

Smart UV project jsem použil pro tvorbu UV mapy stěn fakulty a většiny lightmap. Na rozdíl od UV Unwrapping nevyžaduje Smart UV project označení hran, podle kterých se má objekt „rozbalit“, namísto toho je objekt „rozbalen“ podle nastaveného úhlu maximálního úhlu Angle Limit automaticky. Nástroj automaticky spočítá vhodné rozmístění jednotlivých UV ostrůvků



Obrázek 10: UV Unwrapping

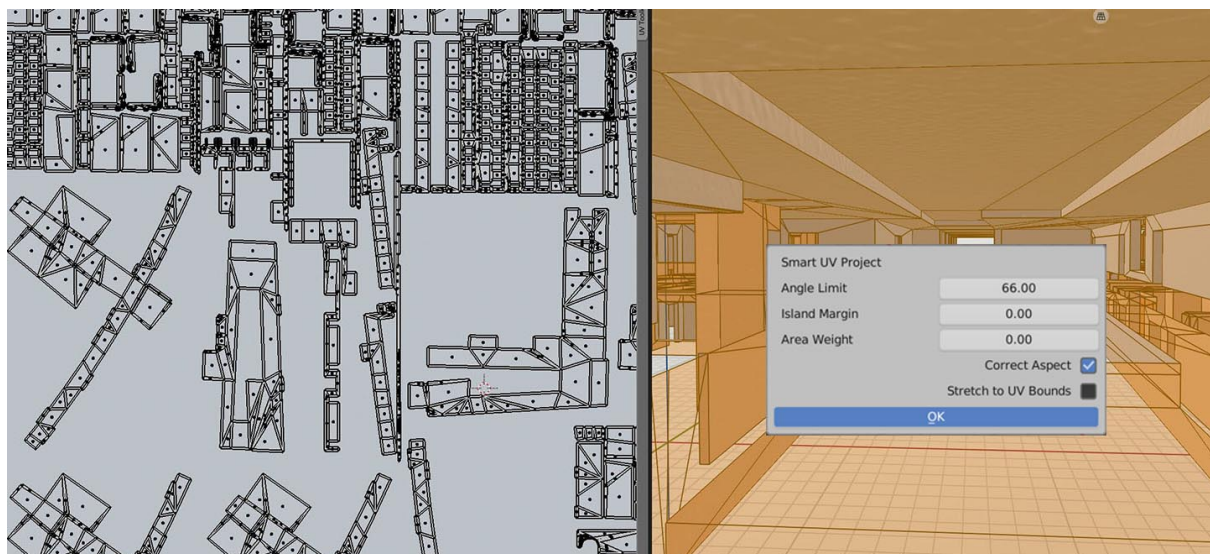
tak, aby co nejlépe pokrývaly UV prostor. Pro tvorbu lightmap do druhého UV kanálu jsem využíval i nastavení Island Margin pro stanovení minimální vzdálenosti mezi jednotlivými UV ostrůvky. Po vygenerování UV mapy pro účel textury následovalo už jen ruční zvětšení celé vygenerované mapy za hranice UV prostoru tak, aby měl materiál vhodnou hustotu texelů vzhledem k rozměrům objektu. Ukázka na obrázku 11.

Manuální mapování

Pod tímto pojmem je zařazeno více postupů se společným prvkem manuálního „napasování“ UV mapy na předem připravený atlas (atlas je popsán v kapitole 3.2.1). Jako ukázka pro tento postup byla zvolena zadní budova fakulty. Nejprve byl uplatněn postup UV Unwrapping, jak je popsáno výše, což automaticky vytvořilo UV ostrůvky tak, aby co nejvíce pokryly plochu. Dalším krokem bylo napasování jednotlivých hran a vertexů UV mapy na atlas tak, aby odpovídaly textuře atlasu (viz obrázek 12). Velmi užitečný pro zarovnání částí UV mapy (podle osy x nebo y je nástroj Align, který byl ve velké míře použit právě pro napasování UV mapy na panely tvořící stěny zadní části budovy fakulty.

2.4 Příprava pro export do FBX

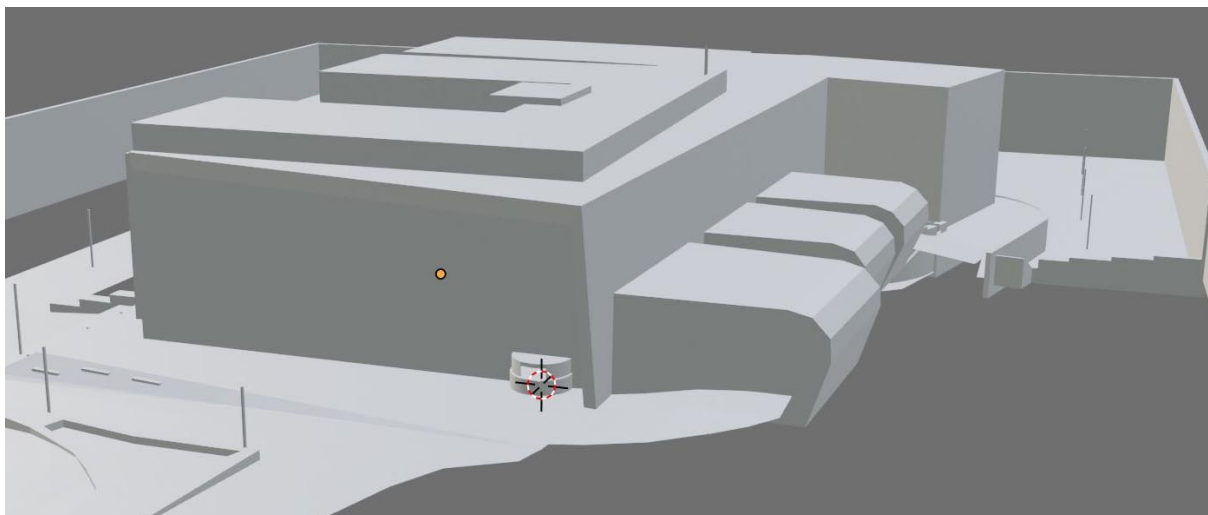
V předchozích sekcích jsem nastínil problematiku modelování, tvorby textur a UV mapování. Tím bohužel příprava pro export do Unreal Engine 4 nekončí. Před exportem bylo potřeba zkontrolovat a nastavit spoustu věcí. Jednotlivé objekty ve scéně bylo třeba vhodně pojmenovat, aby se s nimi dobře pracovalo v Unreal Engine. U každého objektu bylo třeba zkontrolovat správné otočení normál, k čemuž v Blenderu posloužila funkce Face Orientation, která zobrazí správně otočené normály modře a obrácené normály červeně. Dalším krokem bylo aplikovat všechny transformace u jednotlivých objektů a použít u každého objektu nástroje pro pročištění chyb



Obrázek 11: Smart UV Project



Obrázek 12: Manuální mapování



Obrázek 13: Kolizní objekty

modelů, konkrétně Merge By Distance a Degenerate Dissolve. V této části také byly vytvořeny zjednodušené modely objektů ve scéně, které budou následně v Unreal Engineu použity jako kolizní objekty viz obrázek 13

Posledním krokem bylo vytvoření chybějících UV map pro ukládání lightmap. Také bylo nutné zkontrolovat, jestli u těchto UV map nedochází k překrytí a jestli je prostor mezi UV ostrůvky dostatečně velký (aby se zamezilo přetékání světelné informace na okolní ostrůvky). K výběru částí UV mapy, které se překrývají, slouží nástroj Select Overlap. Pro přerozdělení ostrůvků s nastavitelnou minimální vzdáleností mezi ostrůvky slouží funkce Pack Islands.

Pro export do FBX jsem zvolil výchozí nastavení s výjimkou Smoothing, které jsem přenastavil na Edges. Význam tohoto nastavení jsem zjistil v prvním pokusu o import scény do Unreal Engineu. Po dokončení importu Unreal Engine nahlásil upozornění pro všechny objekty, který FBX soubor obsahuje, že nebyla nalezena informace o Smoothing Groups. Takto importované objekty tedy neměly správně nastavené vyhlazování, protože export do fbx nebral zřetel na nastavené ostré hrany.

Výsledný model, tak jak vypadal před exportem do Unreal Engineu, je vidět na obrázku 14.



Obrázek 14: Model před exportem do Unreal Engineu 4

3 Materiály

Materiál definuje typ povrchu, ze kterého je konkrétní předmět vyroben. Můžeme definovat jeho barvu, jak je lesklý, zda je objekt průsvitný či matný můžeme vidět skrz objekt a mnohem více. Technicky řečeno, když světlo ze scény dopadne na povrch, použije se materiál k výpočtu interakce tohoto světla s tímto povrchem.

Tady má být uvedení problematiky materiálů, proč jsou podstatné. Novinky a vývoj.

3.1 Specular vs metallic workflow

V Metallic workflow je hodnota odraznosti pro kovy umístěna do základní barevné mapy spolu s odraženou barvou pro dielektriku. Používá se kovová mapa, která funguje jako maska pro rozlišení kovových a dielektrických dat nalezených v základní barevné mapě. Metallic workflow je pro grafiky jednodušší a méně náchylný k chybám způsobených dodáním nesprávných dielektrických dat. Používá méně texturové paměti a je široce používán například v Unreal Engine 4, Unity, Quixel atd.

Specular workflow je definován pomocí sady map, které jsou dodávány jako textury do vzorkovače v shaderu PBR. PBR je fyzikálně založené renderování (PBR) je metoda stínování a renderování, která poskytuje přesnější reprezentaci interakce světla s povrchy. Označuje se jako fyzikálně založené renderování (PBR) nebo fyzikálně založené stínování (PBS). V závislosti na tom, o kterém postupu se diskutuje, se metoda PBS používá pro stínovací koncepty a metoda PBR se používá pro vykreslování a osvětlení [12]. Mapy specifické pro tento workflow jsou difúzní, zrcadlové a lesklé. Specular workflow využívá většina špičkových 3D renderů. Několik příkladů zahrnuje Vray, Corona, Renderman, Arnold.

3.1.1 Materiály v Unreal Engine

Materiál je asset, který lze aplikovat na meshe pro kontrolu vizuálního vzhledu scény. Doslova definuje typ povrchu, ze kterého je předmět vyroben.[13]

- Blend Modes popisují, jak se výstup aktuálního materiálu smísí s tím, co je vykreslováno na pozadí
- Fresnel je termín popisující, jak je viditelné světlo odráženo s různou intenzitou v závislosti na úhlu pohledu
- Refraction je termín používaný pro popis změny směru světla vlivem přenosového média
- Nasvícení průhlednosti

Volumetric Non Directional - Objekt je nasvícen stejným způsobem, který je použit pro vytvoření volumetrického efektu, jakým je například kouř. Bude zobrazeno pouze difúzní světlo bez použití normály

Volumetric Directional - Objekt je nasvícen stejným způsobem, který je použit pro vytvoření volumetrického efektu, jakým je například kouř, ale na rozdíl od předchozího případu normála ovlivňuje nasvícení

Surface - Objekt je nasvícen a přijímá obraz, který je však omezený a ne tak dobrý jako u neprůhledného materiálu [13]

3.2 Příprava textur

Programy, ve kterých se dají tvořit textury, jsou např. GIMP, Crazy Bump, Corel Photopaint a Adobe Photoshop. V mojí práci jsem si zvolil tvorbu textur v programu Adobe Photoshop [14].

3.2.1 Atlas vs seamless texturey

Textura seamless zaručuje, že při opakování textury na modelu nejsou vidět žádné švy. Textura atlas je obrázkem skládajícím se z několika menších obrázků zabalených dohromady pro zmenšení celkového rozměru. Obrázky mohou mít různý rozměr. Ve své práci jsem použil oba typy textur.

3.2.2 Příprava textur a materiálů

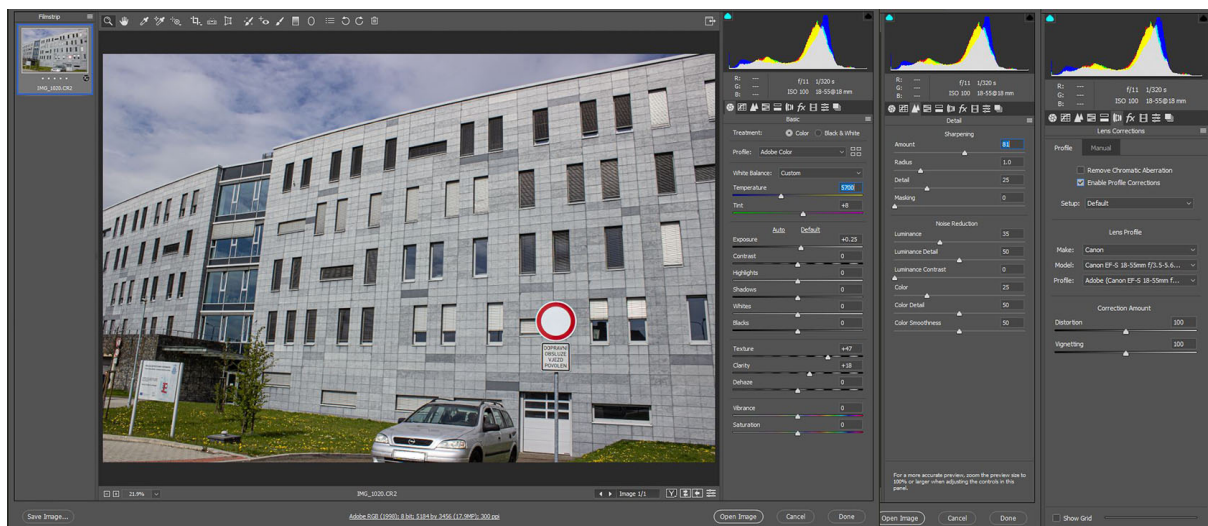
V této podkapitole stručně popíši a předvedu na ukázkách tvorbu jednotlivých textur pro výsledný PBR materiál v programu Adobe Photoshop. Prvním krokem bylo vytvoření prázdného souboru o rozměrech 4192x4192 pro bezešvé (seamless) textury a 8192x8192 pro atlasy. Rozměr pro atlas byl zvolen podle maximálního rozlišení, se kterým Unreal Engine 4 umí pracovat. Rozlišení pro bezešvé textury jsem zvolil pro zachování co největšího množství detailů z fotografií ve formátu RAW. Dalším krokem bylo postupné vkládání fotografií ve formátu RAW a jejich umístění na příhodné místo v atlasu tak, aby byla plocha textury co nejlépe využita. Jako ukázkou jsem zvolil tvorbu atlasu exteriorAtlas 1, kterým je texturována většina částí exteriéru fakulty.

Nastavení vkládaných RAW obrázků

U většiny fotografií jsem při importu obrázku ve formátu RAW do Photoshopu upravoval expozici, aby byla fotka vhodně nasvícená, dále jsem použil Texture a clarity, což zvýrazní detaily na fotografii a používá se často pro tvorbu textur do her. V další záložce jsem podle kvality fotografie upravoval přístřežení (sharpening) a snížení množství šumu (noise reduction). Poslední velmi podstatnou úpravou je korekce zakřivení čočky, ve které musíme nastavit profil pro čočku fotoaparátu, kterým byly fotografie pořizovány. Příklad těchto úprav znázorňuji na obrázku 15.

3.2.3 Tvorba Base Color mapy

Jako první krok po vložení fotografie do atlasu je srovnání perspektivy pomocí nástroje Perspective Warp. Druhým nejpodstatnějším nástrojem je klonovací razítko na zakrytí objektů,



Obrázek 15: Nastavení formátu RAW

které na textuře nechceme, a doplnění částí textury, která na fotce chybí. V případě ukázky by se jednalo o zakrytí dopravní značky Zákaz vjezdu automobilů. Dalšími používaným nástroji jsou například výběr, magická hůlka, guma, štětec, atd.

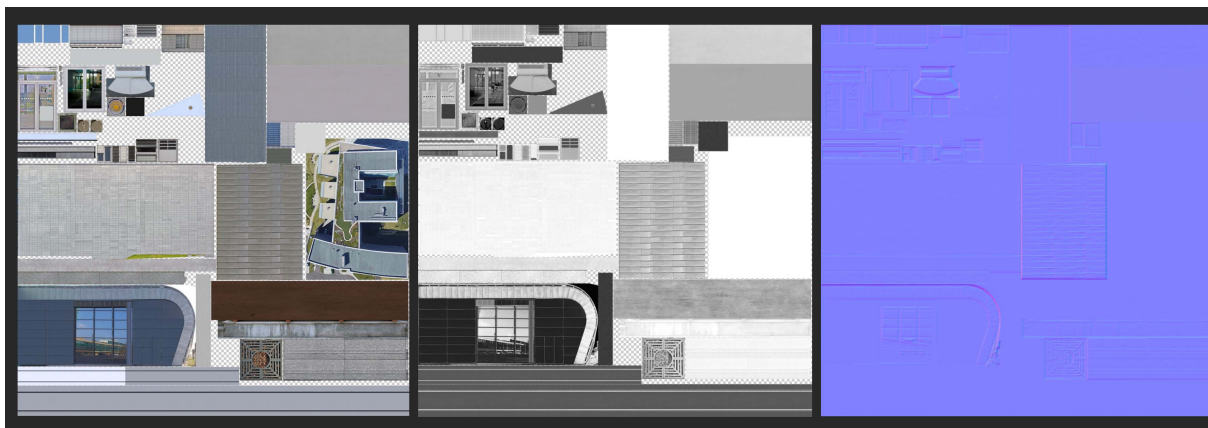
Poté, co je už atlas skoro připraven, je třeba provést barevnou korekci pomocí nástrojů: Color balance, Vibrance, Hue and Saturation. U Metallic postupu také nechceme, aby barevná mapa obsahovala příliš mnoho stínů, pro jejich zmírnění použijeme nástroj Shadows and Highlights.

Tvorba Roughness mapy

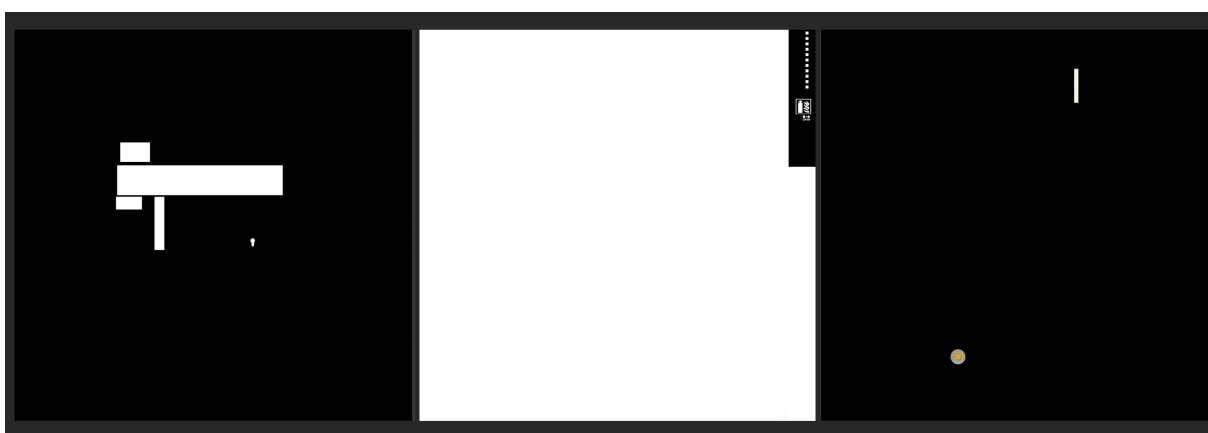
Pro vytvoření Roughness mapy jednotlivé vrstvy z barevné mapy převedeme do odstínu šedi pomocí filtru Black and White a následně provedeme doladění pomocí nástrojů Curves a Levels tak, aby se části textury, které jsou mají být nejvíce hrubé, blížily co nejvíce bílým odstínům. Hladké části, u kterých požadujeme vyšší odlesk, jako například okna či kovové plochy, aby se blížily černým odstínům. Výsledek mapy hrubosti můžeme vidět v levé části obrázku 16

Tvorba Normal mapy

Vytvoření normálové mapy provedeme z mapy hrubosti, kterou však předtím upravíme tak, aby místa, která mají vystupovat, byla bílá a místa, která mají být zapuštěná, byla černá. Pro vygenerování normálové mapy potom použijeme filter Generate Normal Map. Úroveň detailu normálové mapy můžeme nastavit před vygenerováním posuvníkem. Stejně jako u mapy hrubosti jsem normálovou mapu vytvářel po jednotlivých vrstvách, abych u každé části atlasu měl větší kontrolu nad výsledkem. Hotovou Normal mapu můžeme vidět v pravé části obrázku 16



Obrázek 16: Vlevo Base Color mapa, uprostřed Roughness mapa, vpravo Normal mapa



Obrázek 17: Vlevo Metallic mapa, uprostřed Alpha mapa, vpravo Emission mapa

Další použité mapy

Výše zmíněné tři mapy jsou použité pro všechny materiály. U textur a atlasů, které obsahují jak dielektrické tak kovové části, jsem použil Metallic mapu. Pro maskované materiály jsem použil Alpha mapu. Alpha i Metallic mapy jsou jednoduché na vytvoření, stačí ve Photoshopu vyplnit kovové části bílou barvou a u albedo mapy viditelné části bílou barvou a zbytek ponechat černý. Podobný postup platí i pro Emission mapu s rozdílem, že části, které nevyzařují světlo, jsou černé a části, které vyzařují světlo jsou barevné. Ukázka viz 17

3.3 Použité formáty

Datový formát a jeho případná komprese je parametr týkající se kvality digitálního obrazu. Mezi mnou nejčastěji používanými byly formáty RAW, PSD a JPG.

RAW

RAW de facto není obrazovým formátem, jsou v něm uložená veškerá "surová" data snímače a veškerá nastavení fotoaparátu. Jde o "čistá" nekomprimovaná data. V případě snímání do RAWu firmware přístroje nevytváří reálný obraz, nýbrž pouze uloží na paměťové médium data, která pro vygenerování obrazu slouží. Samotné vytvoření fotografie pak probíhá v počítači. Výhodou je, že uživatelsky lze ovlivnit několik snímacích parametrů i po expozici. Tento formát byl použit pro fotografie určené ke tvorbě textur.

PSD

Soubor PSD je soubor obrázku vytvořený Adobe Photoshopem, profesionálním programem pro úpravy obrázků, který se často používá pro zdokonalení digitálních fotografií a tvorbu webové grafiky. Je to nativní formát používaný pro ukládání souborů ve Photoshopu. Soubory PSD mohou obsahovat obrazové vrstvy, vrstvy úprav, masky vrstev, anotace, informace o souborech, klíčová slova a další prvky specifické pro aplikaci Photoshop. Jsou běžně vytvářeny a sdíleny mezi grafickými profesionály. Výhodou je, že se jedná o bezztrátový formát. Pracuje se v tomto formátu a pak se ukládá do JPEG [15].

JPG

Tento formát, resp. JPEG (The Joint Photographics Experts Group,) je jedním z nejrozšířenějších formátů [16]. Je to datový formát primárně navržený pro ukládání fotografií. Jeho typickou vlastností je ztrátová komprese. Pro JPG (JPEG) je charakteristické, že čím více je obraz ztrátově komprimován, tím má menší datovou velikost, ale také tím více ztrácí svojí původní kvalitu, resp. drobné detaily. Stupeň komprese lze uživatelsky nastavit podle typu focené předlohy. Má ale navrch před ostatními formáty co do výsledné datové velikosti souborů. Tento formát byl použitý pro export výsledných textur kvůli výraznému ušetření místa na pevném disku.

4 Herní engine

Unreal Engine od společnosti Epic Games [17] je jeden z nepoužívanějších nástrojů a nejznámějších enginů pro tvorbu her [18], vizualizací, simulací i filmů či seriálů. Unreal engine byl například použit pro vytvoření zjednodušené real-time verze scén pro osmou sérii seriálu Game of Thrones, ve kterých mohli filmaři ve Virtuální Realitě získat lepší představu o podobě výsledné scény ještě předtím, než byly tyto scény vytvořeny a vyrenderovány ve vysoké kvalitě. Game of Thrones. Velké herní společnosti, jako například Electronic Arts si vyvíjejí svůj vlastní engine, například Frostbite, který tvoří jádro her, jako jsou například Need for Speed, Mass Effect, Fifa či Battlefield.

Pod záštitou Epic Games vznikly v Unreal Engine hry jako Unreal Tournament, série her Harry Potter nebo Borderlands.

Existují konkurenční volně přístupné enginy. Unreal Engine jsem si oproti ostatním zvolil proto, že je přívětivější na naučení a umožňuje používat systém blueprintů, který umožňuje vytvoření jednodušší funkcionality bez nutnosti psaní kódu, i klasické programování v jazyce C++ [13],

Mezi konkurenční enginy patří Unity [19] a Cry Engine 3 [20]. Každý z jmenovaných má své výhody a nevýhody. Uživatelé a grafická studia si je vybírají podle očekávaného výsledku. Unity má například velké množství zdarma přístupných materiálů, modelů a skriptů. Strategická 2D hra s názvem Plague Inc: Evolved byla vytvořena v Unity Engine. Cry Engine je těžší pro naučení se, ale nabízí velmi pokročilé grafické techniky, je uživatelsky příjemnější a má velké množství filmových nástrojů. Mezi nejznámější hry vytvořené v CryEngine patří série Crysis či česká hra Kingdom Come: Deliverance.

Existuje velké množství dostupné literatury a on-line informačních zdrojů, které se věnují problematice Unreal Enginu [21], jeho možnostech či optimalizaci, v posledních třech letech se touto problematikou zabývali i studenti FEI ve svých bakalářských a diplomových pracích [22],[2],[23] [24].

Cílem diplomové práce Nové možnosti Unreal Enginu Martina Mady bylo se zaměřit na možnosti vizualizace v novém Unreal Engine 4. Práce se zabývala popisem a demonstrací vybraných částí tohoto enginu a výsledky byly následně použity ve hře pro zkoumání chování lidí v bludišti.

V bakalářské práci Optimalizační techniky v Unreal Enginu 4 Jan Trubač popisuje optimalizaci 3D scény v realtime počítačové grafice. Práce popisuje základní rozdíly mezi realtime a offline 3D počítačovou grafikou. Pro praktickou demonstraci vytvořil Jan Trubač aplikace 3D vizualizace v architektuře. Výstupem diplomové práce Možnosti Unreal Engine 4 studenta Štěpána Guňi byly zásuvné moduly pro vizualizaci map virtuálního světa a nastavování grafických možností aplikací vytvořených v Unreal Engine 4.

Proč je lepší používat ve filmech počítačem generované scénérie namísto skutečných

4.1 Možnosti osvětlení Unreal Engine 4

Nasvícení scény je to, co udělá z monotonního a nudného místa reálně vypadající místo s atmosférou. V Unreal Engineu používáme různé typy osvětlení.

4.1.1 Pohybovost světél - světla statická, stacionární a pohyblivá

Statická světla jsou světla, která nelze za žádných okolností změnit ani přesunout. Jsou vypočítána pouze v rámci lightmap a po zpracování již nemají žádný další dopad na výkon. Pohyblivé objekty se nemohou integrovat se statickými světly, takže užitečnost statických světél je omezená. Statická světla používají pouze lightmapy, které je nutné před použitím spočítat (Lightmass Bake). To znamená, že nemohou stínovat pohyblivé (dynamické) objekty. Jsou nejméně náročné na výkon v reálném čase, ale jejich upečení může trvat velmi dlouho.

Stacionární světla jsou světla, která zůstávají v jedné poloze, ale může se měnit jejich jas a barva. Toto je primární způsob, kterým se liší od statických světél, která se během hry nemohou nijak změnit. Umožňují částečné pečení světél a dynamické stíny pro stacionární či pohyblivé objekty.

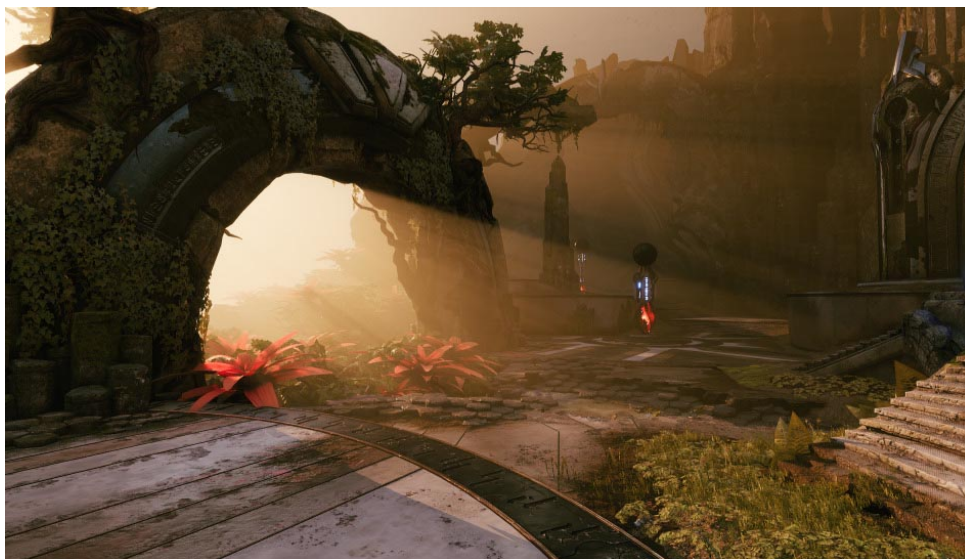
Pohyblivá světla vrhají zcela dynamická světla a stíny. Mohou změnit polohu, rotaci, barvu, jas, falloff, poloměr a téměř všechny ostatní vlastnosti, které mají. Neumožňují pečení světél.

Ve své práci jsem používal převážně světla statická, neboť světla stacionární a pohyblivá měla extrémní dopad na výsledný výkon scény.

4.1.2 Typy světél v Unreal Engine 4

V Unreal Engineu 4 jsou k dispozici tyto typy světél:

- Directional Light – směrové světlo simuluje světlo, které je emitováno ze zdroje, který je nekonečně daleko. To znamená, že všechny stíny vrhané tímto světlem budou rovnoběžné, což z něj činí ideální volbu pro simulaci slunečního světla. Toto světlo jsem použil pro nasvícení exteriérů a interiérů a pro výpočet globálního osvětlení.
- Point Light – bodové světlo. Bodová světla fungují podobně jako žárovka ve skutečném světě a vyzařují světlo ve všech směrech z wolframového vlákna žárovky. Toto světlo jsem nevyužil.
- Spot Light – kuželové světlo. Bodové světlo emituje světlo z jednoho bodu ve tvaru kužele. Použil jsem pro nasvícení interiéru.
- Skylight – ambientní světlo. Zachycuje scénu globálně a aplikuje zachycené světlo na všechny objekty. Toto světlo jsem použil pro nasvícení exteriérů a interiérů a pro výpočet globálního osvětlení.



Obrázek 18: Ukázka volumetrické mlhy z dokumentace Unreal Enginu

- Rectangular - obdélníkové světlo. Emituje světlo do scény z pravoúhlé roviny s definovanou šířkou a výškou. Můžeme je použít k simulaci jakéhokoli zdroje světla, který má obdélníkové oblasti, jako jsou televizory nebo obrazovky monitoru, stropní svítidla nebo nástěnné svítidla. Já jsem je využil na nasvícení bočních chodeb a schodů.

4.1.3 Volumetrická mlha

Volumetrická mlha je efekt simulující prosvítání paprsků slunce atmosférou. Efekt je velice intenzivní a dodává scéně realističnost a atmosféru, je možné ji použít pro mlhu, prach, kouř, aj. Přidáme-li volumetrické světlo, dojde k dalšímu zesílení efektů. Ukázka na obrázku 18

4.1.4 Lightmass globální iluminace

Lightmass vytváří lightmapy se složitými světelnými interakcemi, jako je stínování oblastí a difúzní interreflexce. Používá se k výpočtu globální iluminace. Komunikaci mezi editorem a Lightmassem zajišťuje Swarm Agent, který lokálně řídí sestavení osvětlení a může také distribuovat osvětlení do vzdálených počítačů. Difúzní interreflekce je nejvíce vizuálně důležitým světelným efektem globálního osvětlení. V základním nastavení lightmassy určuje albedo část materiálu, jak moc světla a jaká barva se odráží do okolí.

4.1.5 Lightmass portals

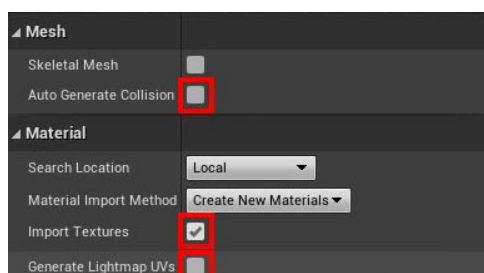
Chceme-li pomoci Lightmass lépe pochopit, odkud by světlo mělo přicházet, můžeme umístit do scény portály Lightmass do oblastí, které jsou pro osvětlení kritické. Lightmass Portals pracují následujícím způsobem. Slouží k vyznačení podstatných částí pro výpočet osvětlení interiéru. Většinou se jedná o okna, či otvory do uzavřených prostor. Když do okna vložíme lightmass

portal, tak určíme oblast, ze které přichází světlo a tedy usnadníme lightmasse výpočet a zvýšíme výslednou kvalitu nasvícení. Lightmass portály je nejvhodnější použít v kombinaci s oblohou.

4.2 Import vytvořeného modelu

Prvním krokem bylo vytvoření nového projektu pojmenovaného jako Faculty. Jako kategorii pro projekt jsem zvolil hry (Games). Jako šablonu jsem zvolil první osobu, jelikož budu používat upravenou verzi Blueprintu FirstPerson pro pohyb po budově a interakci s objekty. Na této volbě příliš nezáleží, protože moduly, které jednotlivé šablony obsahují, se dají přidat i dodatečně. Po načtení nově vytvořeného projektu byly odstraněny všechny objekty ve scéně kromě FirstPersonCharacter, Atmosferic Fog, LightSource a SkyLight, jelikož se s nimi bude dále pracovat. Okna rozhraní byla přerozdělena tak, aby část s náhledem (Viewport) byla co největší. Prohlížeče obsahu (Content Browser) používám dva. Pro přehledné rozdělení prvků, se kterými se bude pracovat, byla vytvořena následující složková struktura

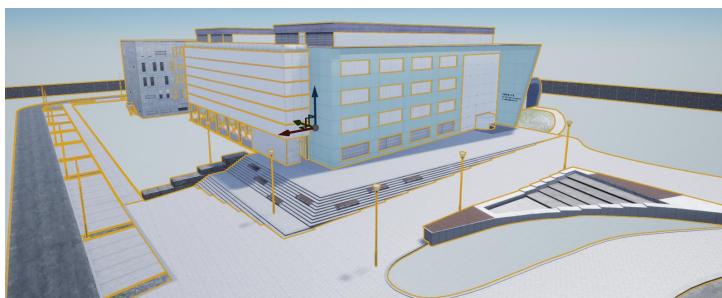
- StaticMeshes - zde jsou ukládány všechny nepohyblivé texturované objekty.
- Materials - zde jsou ukládány všechny materiály a textury
- Colliders - zde jsou ukládány všechny kolizní objekty
- Blueprints - zde jsou ukládány všechny Blueprintsy
- DynamicMeshes - zde jsou ukládány všechny pohyblivé objekty a objekty pro interakci, jakými jsou například: dveře
- Props - pro objekty, které se budou ručně umísťovat do scény. Například: gauče, lavičky, židle, stoly.



Obrázek 19: Nastavení pro import z FBX

Následovalo importování scény z připraveného FBX souboru. Pro import bylo důležité, aby nebyla zatržena volba Generate lightmap UVs, jelikož lightmapy již jsou připravené v druhé UV mapě všech objektů. Dále bylo třeba zrušit volbu Auto Generate Collisions, jelikož budou použity kolizní meshy vytvořené v Blenderu. Pro nastavení materiálů bylo zvoleno vytvoření nových materiálů a automatické importování textur. Zbytek nastavení byl ponechán ve výchozím

stavu. (viz obrázek 19. Importování bylo potvrzeno volbou Import All. Je nutné zmínit, že tímto způsobem importované objekty mají všechny Pivot Point na souřadnicích $x:0$, $y:0$, $z:0$. Proto tento způsob není vhodný pro vložení objektů, se kterými se bude ve scéně manipulovat. Je to však vhodný způsob, když chceme zajistit, aby objekty byly v Unreal Engine na stejné pozici jako ve scéně v Blenderu. Z toho důvodu obsahoval prvotní FBX pouze statické objekty a další objekty jako třeba dveře, gauče, lavičky, židle a podobně byly vloženy později po jednom tak, aby měly Pivot Point na žádaném místě. V případě dveří byl například model v programu Blender posunut tak, aby panty dveří, podle kterých dveře rotují, byly přesně ve středu souřadnic v Blenderu na osách X a Y.



Obrázek 20: Scéna po importu z FBX

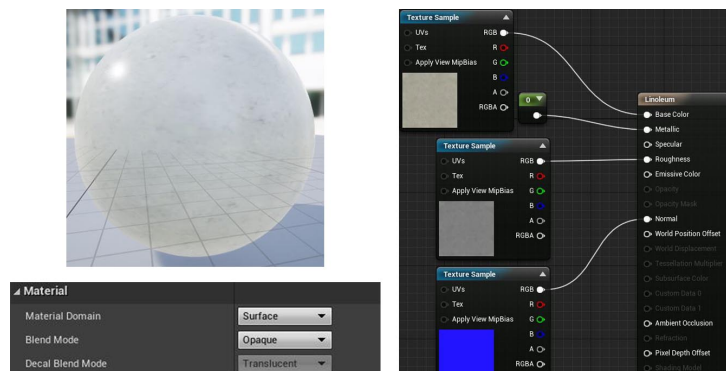
Po dokončení importu byly objekty rozděleny do nachystané složkové struktury. Objekty ze složky Static Meshes byly označeny a přetaženy do 3D pohledu (Viewport). Následně byly vynulovány souřadnice x , y a z . (viz obrázek 20). Po vložení modelu do scény vypadá většina objektů v pořádku a část materiálů se automaticky nastavila správně. Pouze sklo a maskované materiály jsou bílé místo průhledné. Po bližším pozorování si však je možné všimnout, že odlesky materiálů nejsou v pořádku. Převod z Blenderu do Unreal Engine chybně nastavuje na pozici Metallic mapy Roughness mapu. Metallic, Alpha a Emissive textury Unreal Engine vůbec nenaimportoval. Pro účel ukázky nastavení materiálů jsou předvedeny tři případy, a to neprůhledný materiál, průhledný materiál a maskovaný materiál.

4.2.1 Nastavení neprůhledných materiálů

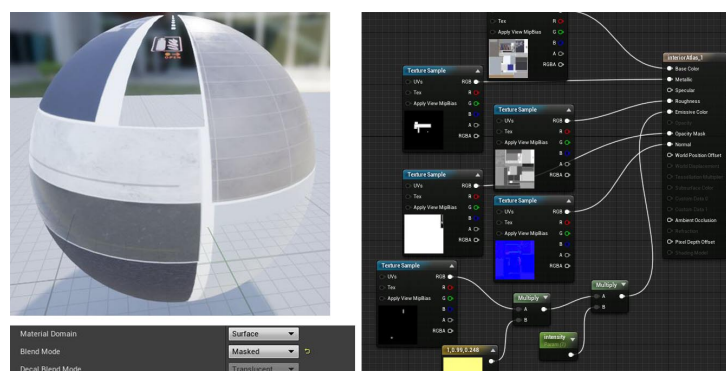
Pro nastavení neprůhledných materiálů slouží Blend mode v režimu Opaque. Na ukázce (viz obrázek 21) byly použity mapy Base Color, Roughness a Normal. V případech, kdy materiál obsahuje dielektrické i kovové část je třeba nastavit Metallic mapu, v případě ukázky je materiál čistě dielektrický a bylo tedy možné nastavit Metallic konstatně na 0.

4.2.2 Nastavení maskovaných materiálů

V tomto případě používáme Blend mode nastavený na Masked. Nachystanou alfa mapu použijeme pro nastavení masky průhlednosti. U tohoto materiálu používáme emisní mapu nastavenou jako Emissive Color. Na obrázku 22



Obrázek 21: Ukázka neprůhledného materiálu linoleum

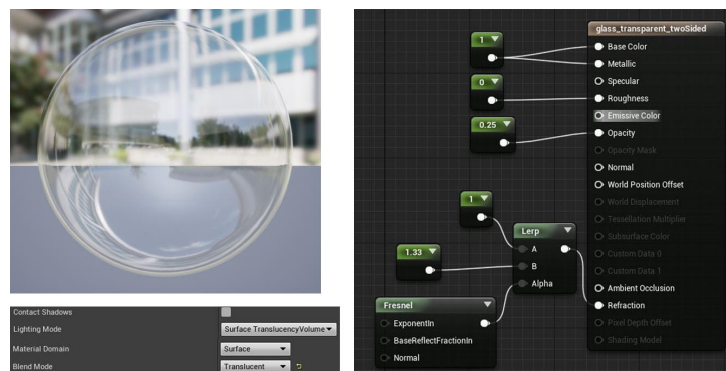


Obrázek 22: Ukázka maskovaného materiálu a použití Emission mapy

je znázorněno použití emisní mapy v kombinaci se složkou multiply. Barevná část emisní mapy nastavená do vstupu A u multiply v tomto případě slouží pro vyznačení částí materiálu, která bude vydávat světlo. Třísložkový vektor nastavený jako vstup B pro multiply násobí toto světlo nastavenou barvou, tedy můžeme takto korigovat výslednou barvu světla. Druhá multiply složka má potom výsledek předchozí složky multiply nastavený jako vstup A a na vstupu B má parametr, který slouží k nastavení intenzity tohoto světla. V pravé části obrázku je ukázka vyzářujícího světla ve scéně.

4.2.3 Nastavení průhledných materiálů

Pro nastavení průhledných skel je použit Blend Mode nastavený na Translucent a jako režim nasvícení průhlednosti je použit Surface Translucency Volume. U tohoto materiálu nepoužíváme žádné texturové mapy, ale vše je nastaveno numericky pomocí konstant. Pro Base Color je použita konstanta 1, pro Roughness 0, jelikož se jedná o dokonale hladký povrch a pro Metallic 1, pro získání barevného odrazu, jako u kovů. Pro simulaci lomu světla je využit Fresnel sloužící jako maska pro lineární interpolaci s lomem světla (Refraction) nastaveného konstantou na 1.52, což je index lomu světla pro sklo. Ukázka na obrázku 23.



Obrázek 23: Ukázka materiálu průhledného skla

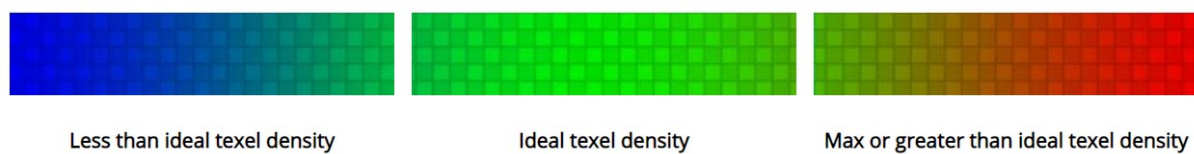
4.3 Nasvícení modelu fakulty

Nasvícení výsledné scény z technického hlediska bylo velmi komplikované. Scéna je poměrně velká s velkým množstvím polygonů a lze v ní bez načítání přecházet z exteriéru do interiéru. Tento případ je jeden z nejtěžších pro real-time grafiku a většinou se neobejde bez triků a chytře použitého postprocessingu. V rámci testování bylo vyzkoušeno velké množství variant. Pro hlavní osvětlení scény slouží směrové světlo simulující slunce a obloha. Tato světla jsou nastavená v režimu Stationary, tedy podporují jak pečené světlo, tak dynamické světlo. Budova fakulty je statická, tedy je nasvícená pouze statickým světlem. Dynamické osvětlení slouží pro stromy v blízkém okolí fakulty.

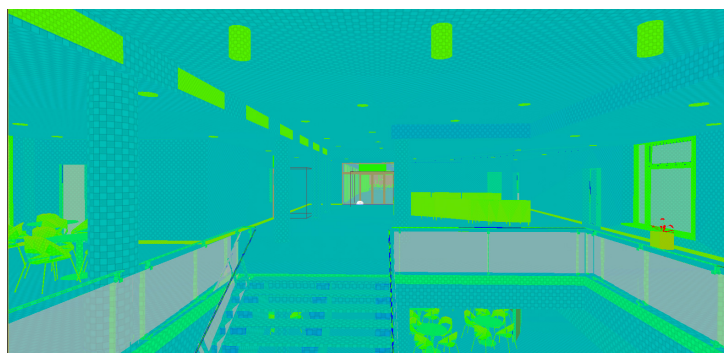
4.3.1 Pečení světla (Lighmass bake)

Pro statické objekty může být světlo spočítáno dopředu a uloženo do připravených lightmap. Před samotným výpočtem světla je však třeba nastavit rozlišení těchto lightmap. Čím vyšší rozlišení, tím lepší kvalita, ale zároveň významně delší doba výpočtu nasvícení a větší nároky na paměť. Vhodným nástrojem 3D pohledu (viewport) je optimalizační zobrazení lightmap Density, které zobrazí hustotu texelů pro všechny objekty ve scéně. Ke správnému nastavení napomáhá barevný tón od modré do červené, jak je znázorněno na obrázku 25. Doporučená hustota je v odstínech zelené. V případě této práce nebylo možné použít doporučenou texelovou hustotu, jelikož při výpočtu nasvícení již nestačilo 32 GB paměti RAM a celý program i počítač se zasekl a veškerý průběh výpočtu nasvícení byl ztracen. Musel jsem se tedy spokojit s nastavením hustoty texelů v odstínech světle modré.

Další komplikací pro výpočet nasvícení byla velmi dlouhá výpočetní doba. Čas výpočtu při nastavení poměrně nízké kvality se pohyboval mezi 2-4 hodinami a během výpočtu byl počítač nepoužitelný pro jinou práci. Výsledný model scény byl počítán asi 8 hodin na procesoru core-i7 8700k.



Obrázek 24: Doporučený texel density z dokumentace Unreal Engine [21]



Obrázek 25: Nastavený texel density pro interiér budovy



Obrázek 26: Nastavení lighmass



Obrázek 27: Vypočtená globální iluminace bez použití dalších světel



Obrázek 28: Vypočtená globální iluminace s přidáním světlů uvnitř budovy

4.3.2 Nastavení Lighmass

Nastavení pro výpočet kvality statického světla se nachází pod World Settings v záložce Lighmass. Zde můžeme nastavit počet odrazů pro nepřímé světlo a oblohu, kvalitu odraženého světla a také hladkost (smoothness) výsledného světla uloženého do lightmap. Nastavení, které jsem použil pro výslednou scénu, je na obrázku 26. Ukázka výpočtu nasvícení bez přidaného umělého osvětlení je na obrázku 27.

4.3.3 Umístění světel do interiéru

Scéna nasvícená pouze nepřímým světlem a oblohou byla příliš tmavá. Zvýšením nastavení Indirect Lighting Intesity u oblohy a nepřímého světla na hodnotu větší než jedna by tento problém mohl být částečně řešen, avšak za cenu přesvětlení okolí fakulty. Toto nastavení tedy bylo ponecháno na výchozí hodnotě a pro nasvícení vnitřní části budovy bylo použito velké množství kuželových světel (Spot Light) pro nasvícení hlavní haly a obdélníkových světel (Rect Light) pro boční chodby a schodiště. Ukázka vypočteného světla s přidaným umělým nastavením je na obrázku 28.

4.4 Nastavení odrazů a odlesků

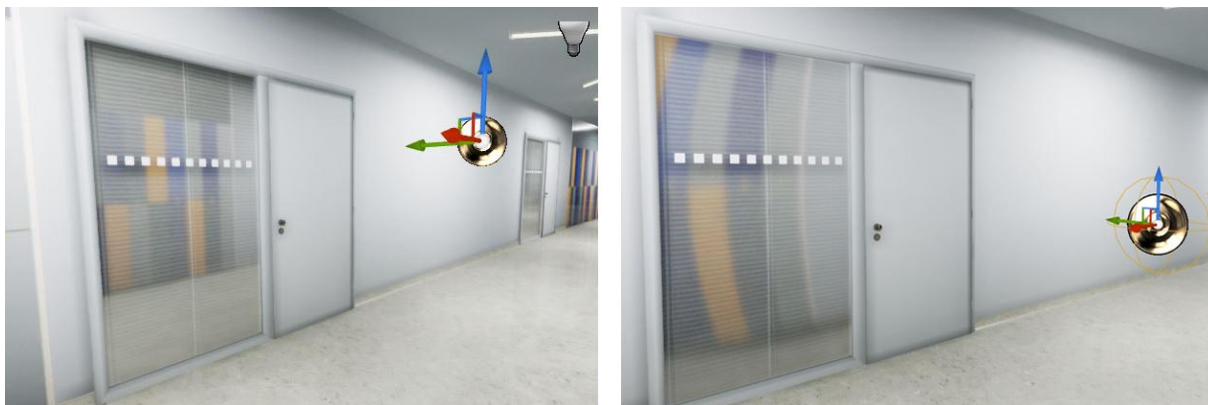
Výpočet odrazu a odlesků okolí je v oblasti real-time grafiky velmi komplikovaný a náročný proces. S příchodem raytracingu s novou generací NVIDIA grafických RTX karet již máme k dispozici skutečné odlesky a odrazy počítané pomocí sledováním paprsků. Ukázka Battlefield V. Jelikož ani já, ani většina lidí ještě nevlastní grafickou kartu s možnostmi RTX, tak jsem v této práci použil starší metody pro odlesky a odrazy. Budova fakulty je plná skleněných a lesklých povrchů, což výrazně zvyšuje časové i technické nároky na správné nastavení odrazů a odlesků. Unreal Engine nabízí několik metod k zachycení odražených objektů.

- Sphere Reflection Capture - slouží k zachycení scény a její projekci na jednoduchý objekt tvaru koule. Získaný odraz je statický a díky tomu i nenáročný na výkon. Vhodné použití pro objekty, na kterých nebude příliš patrné zkreslení.
- Cube Reflection Capture - podobný princip jako Sphere Reflection Capture, V tomto případě je statická scéna zachycena na objekt kvádrů. Vhodné použití je pouze u místností a chodeb tvaru kvádrů, na které je možné Cube Reflection Capture co nejpřesněji naposovat.
- Plane Reflection - nejpřesnější, ale také nejnáročnější způsob řešení odrazů. Umožňuje vykreslit skutečné odrazy v reálném čase podle nastavené roviny. Tento typ řešení odrazů je nutné povolit v nastavení projektu.
- Screen Space Reflections (SSR)- nastavení, které je ve výchozím nastavení povoleno. Je možné ho povolit či zakázat pro jednotlivé materiály. Jak již název napovídá, tak screen space reflections jsou schopny vytvářet skutečný odraz objektů, které jsou aktuálně viditelné na obrazovce. Jakmile se odražený objekt dostane mimo viditelnou oblast, tak už jeho odraz není vykreslován. Vhodným použitím této funkce jsou odrazy postav od lesklé podlahy. Není vhodné používat SSR pro vertikální objekty s velkou odrazivostí, jelikož pak jde velmi zřetelně vidět mizení odlesků jakmile se odražený objekt dostane mimo pohled kamery.

Pro účely této práce byly použity převážně Cube Reflection Capture pro chodby a místnosti interiéru budovy a Sphere Reflection Capture pro zbytek. Ukázka srovnání Cube Reflection Capture (vlevo) a Sphere Reflection Capture (vpravo) na obrázku 29.

4.5 Pohyb ve scéně

Pro pohyb ve scéně pomocí myši a klávesnice, gamepadu či ve VR musí být nastaveno několik základních věcí. Nejprve je třeba vytvořit, nebo použít Blueprint sloužící jako ovládaná postava. Existuje několik základních přístupů pro ovládání postavy. Ve dnešních hrách se používá nejčastěji pohled z první osoby (odkaz) a ze třetí osoby (odkaz). Pohled z první osoby je vhodný zejména pokud je hlavním cílem ztotožnění hráče s postavou či co největší rozhled po okolí. Třetí



Obrázek 29: Porovnání Cube Reflection Capture (vlevo) a Sphere Reflection Capture (vpravo)

osoba je vhodná u příběhových her pro lepší představu a emocích hlavní postavy a úskalích, které používá. Příkladem tohoto využití jsou například série *Uncharted*, *Mass Effect* (odkazy). Dále se používá u bojových her, ve kterých potřebujeme vidět pohyby naší postavy pro správné načasování útoků, skoků a podobně (např: *Dark Souls* (odkaz)). Pro strategie se používá izometrický pohled (<https://www.gamepressure.com/games/pc/strategy/isometric-view/>). Pro účely architektonické vizualizace či filmu nemusí být ovládání postavy vůbec použito, ale namísto toho mohou být použity různé kamery či více kamer pro průlet scénou a přepínání pohledů jak je běžné u filmů.

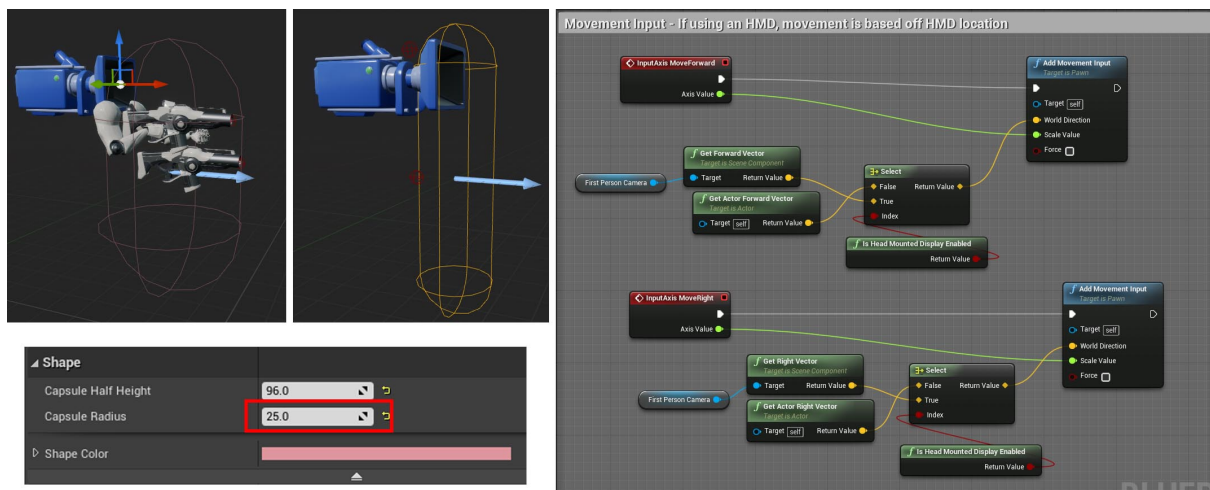
Pro účel této práce nejlépe posloužil pohled z první osoby (tedy i ve virtuální realitě), jelikož podstatnou částí je samotná fakulta a ne ovládaná postava. Další výhodou bylo, že pro postavu z první osoby není na rozdíl od osoby třetí nutné nastavovat žádné animace.

4.5.1 Úprava blueprintu postavy

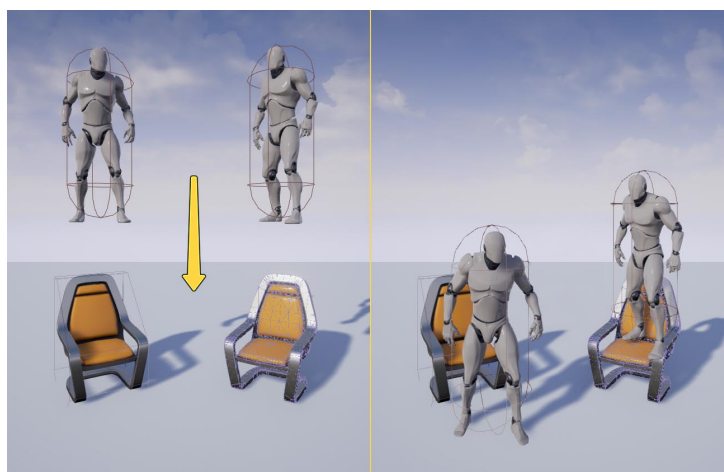
Pro postavu byl použit blueprint `FirstPersonCharacter`, který byl vytvořen zároveň s projektem. Z tohoto blueprintu byly následně odebrány možnosti střelby i zbraň, kterou tento blueprint používá ve výchozím nastavení. Dále byla posunuta kamera více do středu pro lepší pocit při otáčení kamery. Oblast vlivu pro kolize této postavy je určena objektem `Capsule`. Výchozí nastavení mělo příliš velkou hodnotu `Radius`, který neumožňoval projít menšími dveřmi, proto byla tato hodnota změněna na 25. Popsané změny jsou znázorněny na obrázku 30.

4.5.2 Kolize

Aby se mohla postava po scéně pohybovat a aby fungovaly případné fyzikální simulace, je nutné mít nastavené kolize pro jednotlivé objekty ve scéně. Unreal Engine používá jednoduché a komplexní kolize. Na obrázku ?? je srovnání jednoduché a komplexní kolize nastavené na objektu židle. V případě jednoduché kolize postava sjede po nakloněné rovině (která vznikla zjednodu-



Obrázek 30: Upravený blueprint FirstPersonCharacter

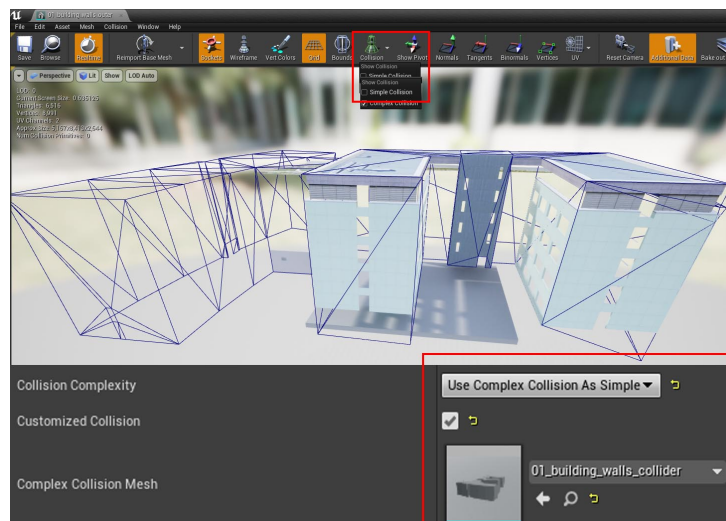


Obrázek 31: Ukázka jednoduché a složité kolize z webu UnrealEngine.com

šením opěradla a sedadla židle), v druhém případě postava zůstane stát na židli s komplexními kolizemi.

- Jednoduché kolize - zjednodušují objekt na primitivum a jsou tedy vhodné pouze pro jednoduché objekty bez vystupujících částí a pro objekty, kterými se nebude procházet. Tento typ je možné použít u dynamických objektů pro simulace.
- Komplexní kolize - používají ve výchozím nastavení kompletní polygonální strukturu objektu a jejich náročnost na výpočet tedy stoupá s komplexností modelu. Řešením toho problému je použití vlastního meshe pro komplexní kolize. Tento typ kolizí lze použít jen pro statické objekty.

V této práci jsou použity meshe vytvořené zjednodušením objektů v programu Blender pro účely komplexních kolizí. Na obrázku 32 je znázorněno použití kolizního meshe pro venkovní



Obrázek 32: Ukázka kolizí

stěny modelu fakulty. Je zde dobře vidět, že kolizní mesh nemusí úplně odpovídat modelu, pro který je použit. V tomto případě obsahuje i část kolizí pro zadní budovu fakulty, která však není součástí tohoto objektu. Zadní budova již tedy nebude mít nastaveny žádné kolize.

4.5.3 Navmesh

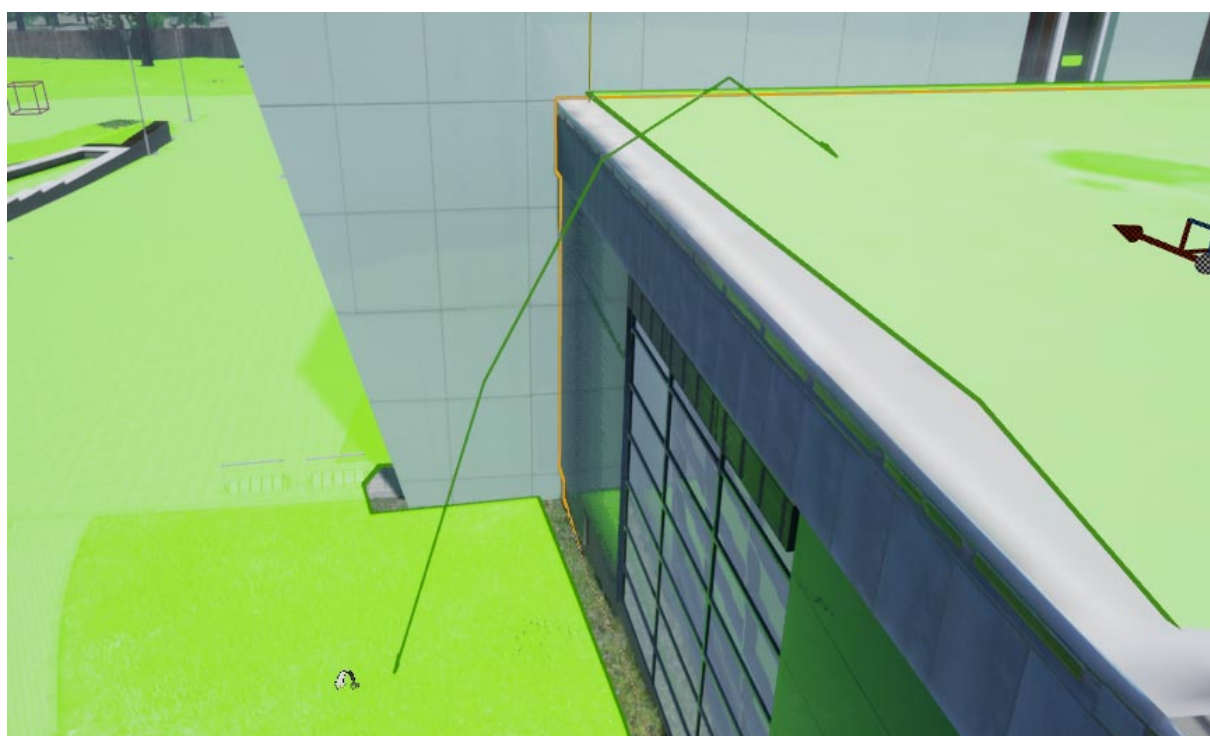
Navmesh slouží pro pohyb postav ovládaných umělou inteligencí (NPC). K vyznačení oblasti, pro kterou se má generovat Navmesh slouží komponent NavMeshBoundsVolume. Ten podle nastavených kolizních meshů vygeneruje cesty, po kterých se můžou NPC pohybovat. Ve Viewport části je možné zkratkou P tyto cesty zobrazit. Vygenerovanou oblast je dále možné modifikovat komponentem Nav Modifier Volume. Například je možné vyznačit oblast, do které nemůžou NPC nikdy vstoupit, nebo oblast, kterou mohou použít pouze v případě, pokud není k dispozici jiná cesta. Na obrázku 33 jsou zeleně vidět cesty pro navigaci NPC. Kolem červeně označené části je umístěn Area Modifier Volume s nastavením NavArea Obstacle, tedy tímto vchodem budou NPC moci chodit pouze tehdy, pokud nebude jiná cesta do vnitřní části budovy. Poslední komponentou je Nav Link Proxy, který slouží k tomu, aby NPC mohli seskočit z jedné části navmeshe na jinou nepropojenou část navmeshe (či vyskočit v opačném případě). Lehce nerealistickou ukázkou tohoto případu můžeme vidět na obrázku 34. S tímto nastavením by mohli aktéři vyskakovat z chodníku na posluchárnu nebo seskakovat z posluchárny.

4.6 Vykreslování terénu a vegetace

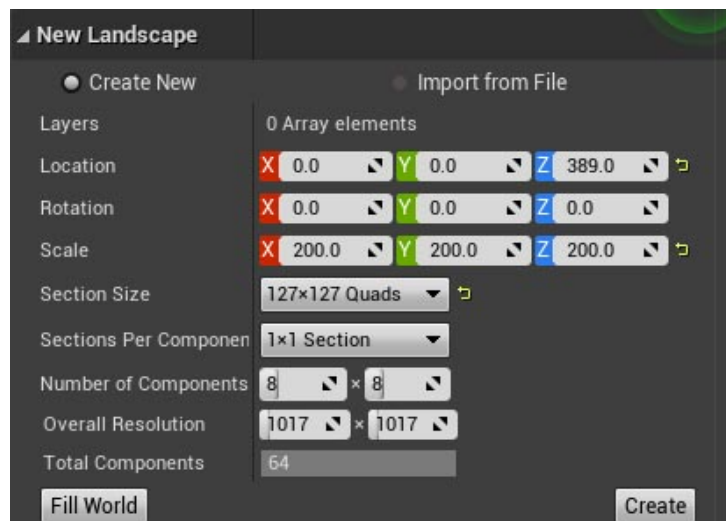
Pro tvorbu terénu slouží v Unreal Engine nástroj Landscape. Terén je možné vytvořit dvěma způsoby. Prvním způsobem je vytvoření nového terénu, což vytvoří rovnou plochu rozčleněnou na komponenty a ty se dále dělí na sekce. Jednotlivé komponenty terénu jsou vždy o stejné



Obrázek 33: Ukázka NavMeshBoundsVolume a Nav Modifier Volume



Obrázek 34: Ukázka Nav Link Proxy



Obrázek 35: Nastavení pro tvorbu terénu

velikosti a mají tvar čtverce. Každá komponenta používá vlastní texturu pro uložení informace o výšce terénu. Tvorba terénu nabízí tyto nastavení.

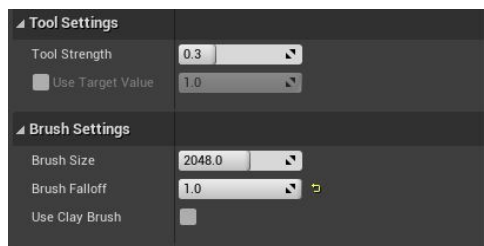
- **Section Size** - nastavení určující velikost sekce, neboli z kolika čtverců se tato sekce skládá. U menších sekcí se agresivněji používá technika úrovně detailu (Level of Detail), což usnadní vykreslování na straně grafické karty, ale za cenu většího využití procesoru. Toto nastavení má vliv na výsledné rozměry terénu, ne na jeho detailnost.
- **Sections Per Component** - Určuje na kolik sekcí je jeden komponent rozdělen. Zdvojnásobením tohoto nastavení dostaneme dvakrát větší terén
- **Number of Components** - Celkový počet komponent, ze kterých se bude skládat výsledný terén

Všechny výše zmíněná nastavení se týkají pouze optimalizace a velikosti výsledného terénu. Jediné nastavení, kterým můžeme ovlivnit detailnost výsledného terénu, je **Scale** (rozměr). Po vytvoření terénu již není možné tato nastavení upravovat. V mojí práci jsem terén nastavil způsobem ukázaným na obrázku 35. Problémem bylo, že jsem špatně pochopil nastavení pro tvorbu terénu a nastavil velikost na 200 pro všechny tři osy. Díky tomu jsem dostal méně detailní terén. Poté, co jsem strávil několik hodin jeho úpravou, aby správně navazoval na chodníky a silnici, už nebylo praktické jej vytvářet znovu.

4.6.1 Úprava terénu

Pro úpravu vytvořené plochy terénu nabízí Unreal Engine několik nástrojů. Všechny tyto nástroje jsou aplikovány přímo na terén pomocí štětce. Nastavení pro štětec je na obrázku 36).

- **Brush Size** - velikost štětce.



Obrázek 36: Nastavení štětce



Obrázek 37: Upravený terén

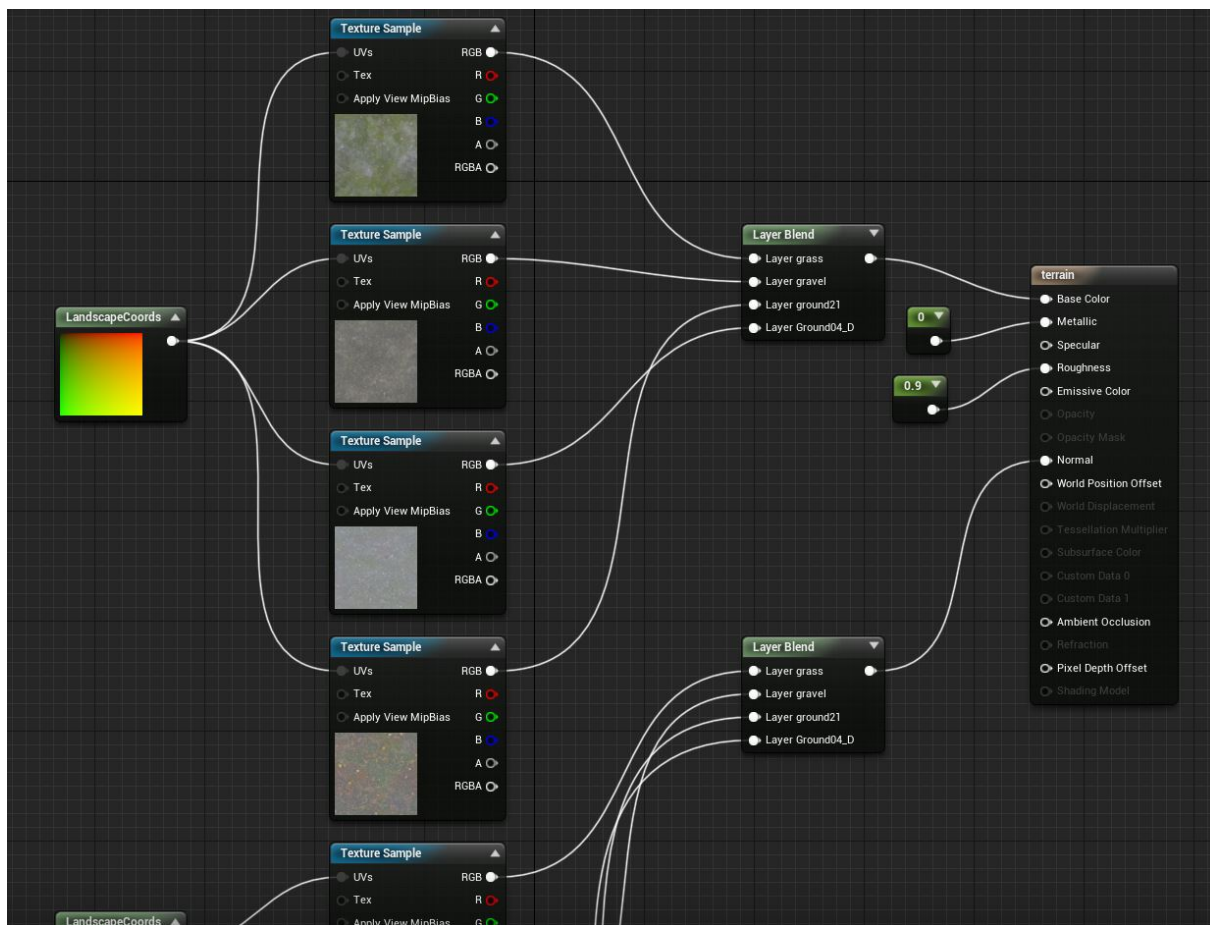
- Falloff - lineární úbytek intenzity působení štětce, hodnoty od 0 do 1. 0 znamená žádný úbytek intenzity, 1 znamená úbytek intenzity od středu k okraji štětce. Velmi užitečné nastavení, při vytváření ostrých přechodů a kontur je vhodné zvolit nízkou hodnotu kolem 0, pro hladké přechody hodnoty mezi 0,5 - 1.
- Tool strength - Síla působení zvoleného nástroje. Pro drobnější změny terénu je vhodné ji snížit.

K úpravě terénu a jeho napasování na obrubníky chodníků a silnice jsem používal převážně nástroj Sculpt, který slouží k vytažení či zapuštění terénu. Tímto nástrojem jsem vytvořil všechny vzdálené kopce a hrubý tvar terénu. Pro doladění výsledku posloužily nástroje Smooth, který vyhlazuje oblast, na kterou je použit a Flatten, který zarovná oblast, na kterou je aplikován, do roviny. Postupnou kombinací těchto tří postupů vznikl terén znázorněný na obrázku 37

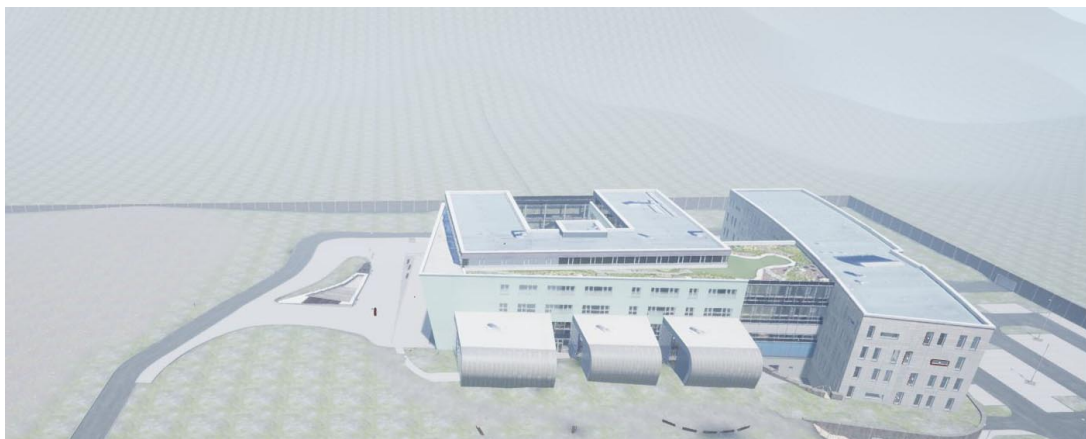
4.6.2 Tvorba materiálu pro terén

K nanesení materiálů na terén slouží nástroj Paint. Tímto nástrojem je možné kreslit na terén více různých povrchů. Aby bylo možné tento nástroj použít, tak je nejprve nutné vytvořit vícevrstevný materiál. Na obrázku 38 je ukázaný materiál použitý pro tuto scénu. Materiál používá tyto uzly

- Landscape Layer Coords Node - slouží pro vygenerování UV souřadnic pro namapování materiálu na terén.



Obrázek 38: Vícevrstevný materiál terénu



Obrázek 39: Terén s materiálem

- Landscape Layer Blend - umožňuje míchání materiálu, aby mohly být použity pro vrstvy terénu. Pro ukládání informací o jednotlivých vrstvách používá pole. Existují tři typy míchání materiálů.

Weight Blend - míchání založené na intenzitě. Je vhodné tehdy, když je vyžadováno nanášení vrstev nezávisle na ostatních vrstvách a jejich pořadí.

Alpha Blend - míchání založené na alfa kanálu. Vhodné použití je pro detaily nebo když chceme mít jasně určené pořadí vrstev. Například sníh zakrývající trávník.

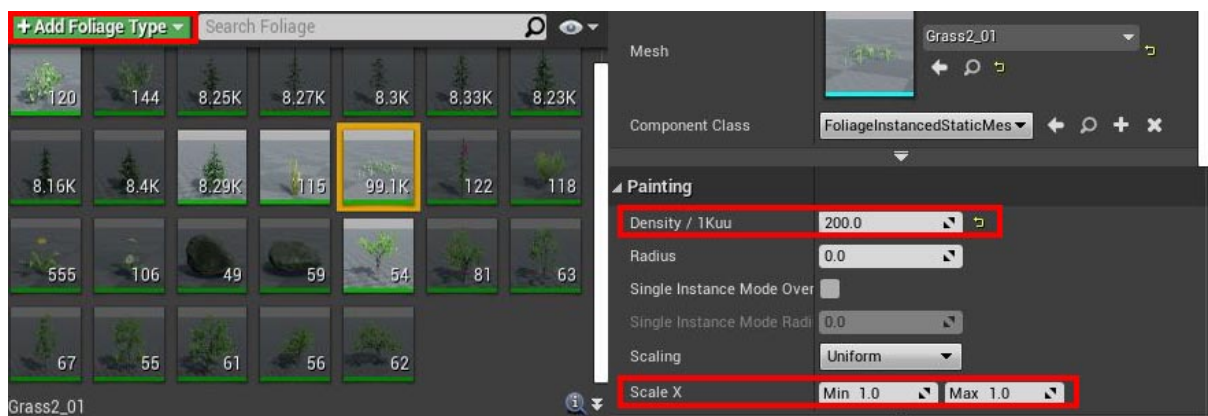
Height Blend - míchání podobné Weight Blend, ale navíc nabízí lepší přechody mezi vrstvami při použití Height mapy (výšková mapa).

Jelikož jsem neměl k dispozici výškové mapy, tak jsem pro materiál terénu zvolil Weight Blend (viz obrázek 39). Po vytvoření byl materiál přiřazen do terénu. Před nanášením materiálu na terén ještě zbývalo vygenerovat Layer Info v režimu Weight-Blended.

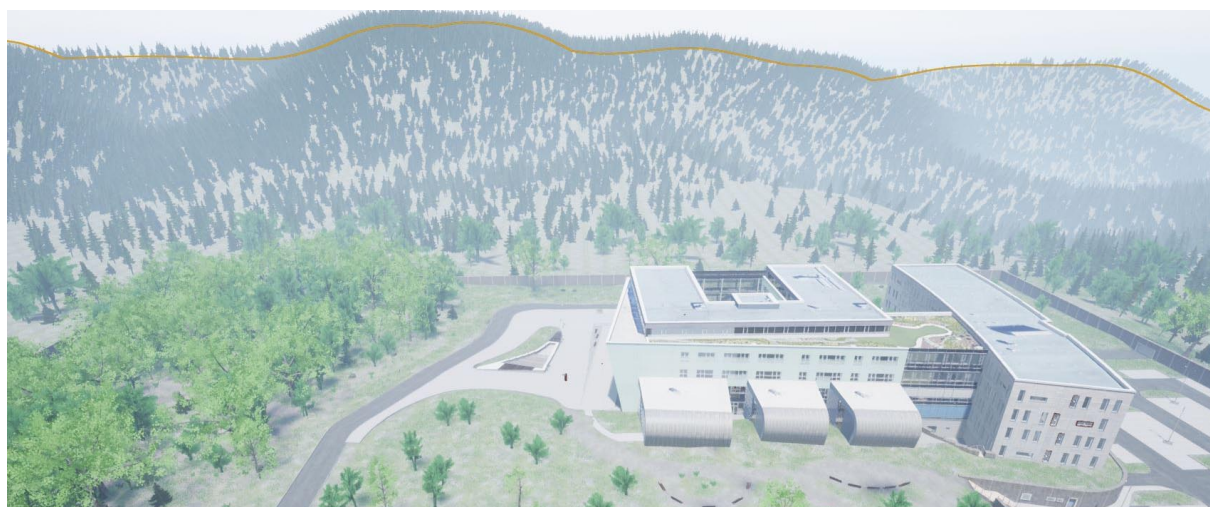
Pro samotné nanášení materiálu na terén slouží štětec se stejným nastavením jako u úpravy terénu (obrázek 36). Terén po nanášení materiálu je zobrazen na obrázku 39. Na tomto obrázku je velmi zřetelné opakování motivu textury trávníku. Po přidání vegetace a při pohledu na scénu z úrovně chodce však již tento problém nebude zřetelný.

4.6.3 Přidání vegetace

Příprava vegetace, tedy od trávy, květin, drobných keřů až po stromy je velmi náročný proces, naštěstí existuje spousta kvalitních zdrojů, ze kterých je možné vegetaci pro scénu získat. Například ze stránky Quixel Megascans. Pro vegetaci v mé scéně jsem použil Nature Package, který byl zdarma dostupný v Unreal Engine Marketplace v dubnu 2020. Pro přidání vegetace slouží nástroj Foliage. Pro nanášení vegetace na terén nebo jejího smazání je použit nástroj Brush. Tento proces je poměrně jednoduchý, stačí přidat typ vegetace tlačítkem Add vegetation, nastavit hustotu pokrytí pro jednotlivé prvky vegetace. Aktivní materiály jsou poté štětcem nanášeny



Obrázek 40: Nastavení pro přidání vegetace



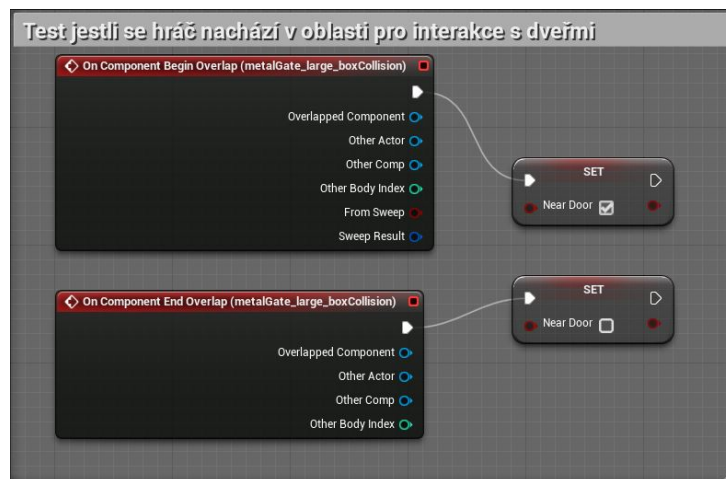
Obrázek 41: Výsledný terén

na terén podobně jako materiál. Tímto způsobem je tedy možné nanášet více typů vegetace naráz. Dalším užitečným nastavením je minimální a maximální velikost. Tímto způsobem můžeme například vytvořit trávník s různou velikostí trsů trávy. Ukázka na obrázku 40.

Z pohledu optimalizace je vegetace velmi náročná na vykreslení, obzvláště pokud vrhá dynamické stíny. Z tohoto důvodu byla vypnuta volba Cast Shadow u všech objektů vegetace kromě stromů v oblasti fakulty ohraničené zídkou. Ukázka hotového terénu po přidání vegetace je na obrázku 39.

4.7 Volumetrická mlha

Unreal Engine nabízí 2 nástroje pro práci s mlhou. Prvním je Atmospheric Fog.



Obrázek 42: Test jestli se hráč nachází v oblasti kolizního boxu

4.8 Interakce se scénou

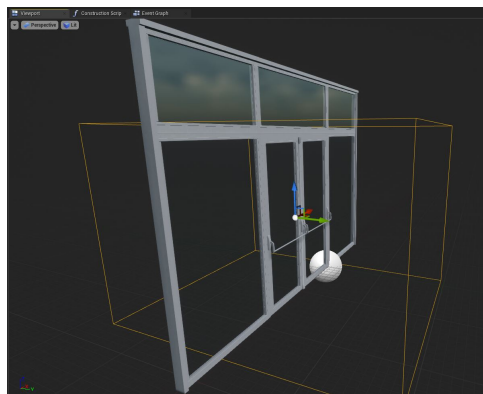
Pro vytvoření interakce se scénou, konkrétně pro vytvoření otevíratelných dveří a plnění úkolů, byl použit systém Blueprints.

Blueprints je kompletní herní skriptovací systém založený na konceptu použití rozhraní založeného na uzlech k vytvoření herních prvků. Stejně jako u mnoha běžných skriptovacích jazyků se používá k definování objektově orientovaných tříd nebo objektů v enginu. Tento systém je mimořádně flexibilní a výkonný, protože umožňuje využívat prakticky celou řadu konceptů a nástrojů, které jsou obecně dostupné pouze programátorům [13].

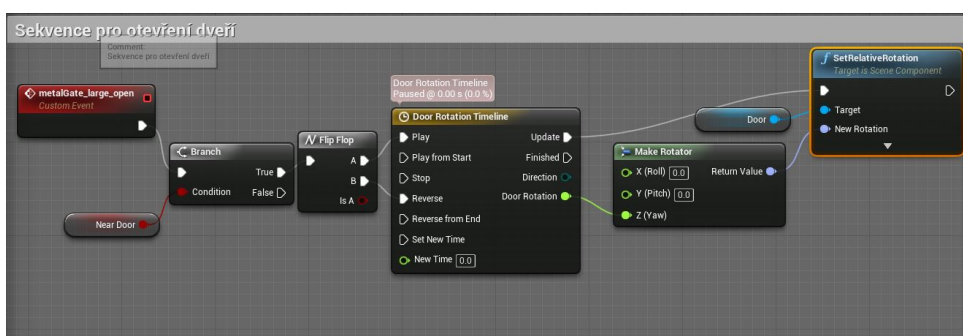
4.8.1 Blueprint otevírání dveří

Objekty dveří, které bude možné otevírat, byly uloženy v samostatných souborech a umístěny v Blender scéně tak, aby panty dveří, kolem kterých bude prováděna rotace, byly ve středu souřadnic. Dalším krokem bylo vytvořit blueprint, který bude použit pro různé typy dveří. Nejprve byl do viewport sekce blueprintu přetažen rám dveří a pod něj byl jako child objekt umístěn objekt dveří. Rám dveří je v blueprintu potřebný k tomu, aby bylo možné s celými dveřmi pohybovat a tedy využít kopie stejného blueprintu pro všechny identické dveře. Následně byl do viewportu vložen objekt boxCollision, který bude sloužit pro vyznačení oblasti, ve které se musí hráč nacházet, aby mohl dveře otevřít. Ukázka dveří s boxCollision je na obrázku 43 a ukázka detekce překrytí (tedy jestli se hráč nachází v oblasti působení boxCollision) je na obrázku 42. Když hráč vstoupí do oblasti vlivu BoxCollision, tak se nastaví proměnná NearDoor na hodnotu true, a když hráč tuto oblast opustí, tak se proměnná NearDoor nastaví na hodnotu false.

Dále byla vytvořena událost (Event) sloužící pro samotné otevření dveří. Nejprve je testována proměnná NearDoor. Pokud je nastavená na hodnotu true, tak se spustí dvousekundová animace přechodu z hodnoty 0 do 90 stupňů. Prvek Flip Flop je použit k přepínání animace, což umožňuje



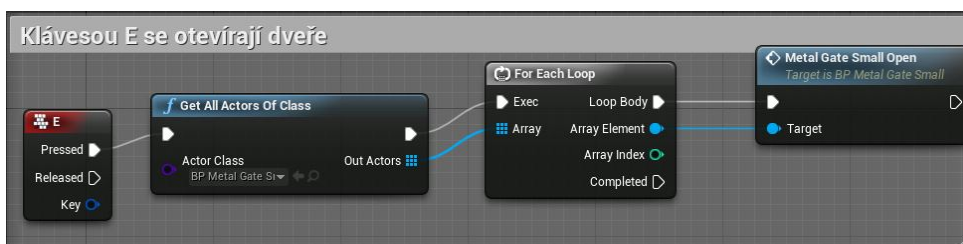
Obrázek 43: Ukázka dveří a zónou ve které je možné dveře otevřít



Obrázek 44: Sekvence otevírání dveří

přerušit animaci v průběhu a otočit její průběh (tedy možnost dveře začít zavírat v době, kdy ještě probíhá animace jejich otevření a naopak). Prvek Make Rotator má přiřazenou animaci otevření dveří pro osu Z a vrací hodnotu této rotace, kterou aplikujeme na objekt dveří. Ukázka tohoto postupu je na obrázku 44.

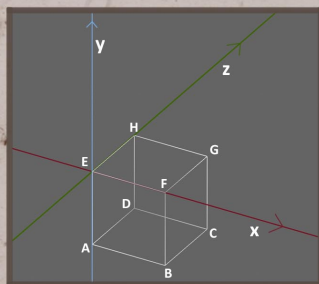
Aby mohl tento blueprint fungovat, je ještě potřeba jej umístit do scény a přidat do Level Blueprintu možnost otevření dveří klávesou E, jak je ukázané na obrázku 45.



Obrázek 45: Detekce interakce otevření dveří

ÚKOL 1

Proved'te rotaci o 180 stupňů podle osy x a translaci o vektor $(0, -2, 12)$ znázorněné krychle (souřadnice bodu A je $(0, -3, 0)$) o délce hrany 3 (jednotka není podstatná). Výsledná souřadnice $X Y Z$ bodu C určuje název místnosti, ve které najdete zadání dalšího úkolu.



Obrázek 46: Ukázka úkolu

4.8.2 Plnění úkolů

Jako jedno z možností využití této práce bylo plnění úkolů souvisejících s výukou různých předmětů. Tato funkcionalita nebyla z časových důvodů úplně dokončena, ale byl vytvořen příklad, jak by takový systém mohl v praxi vypadat a být rozšířen v navazujícím studiu či v projektech jiných studentů. Úkoly byly vytvořeny z vyučované látky předmětu Základy počítačové grafiky (ZPG). Na videu přiloženém k této práci je předveden příchod do budovy fakulty a průchod budovou podle zadaných úkolů na informačních tabulích viditelných ve hře. Ukázka jednoho z úkolů je na obrázku 51.

4.9 Srovnání podobných prací

Problematikou modelování budov se již v minulosti zabývalo několik studentů u nás i v zahraničí. Pro srovnání uvádím dva příklady prací studentů, kteří tvořili model a vnitřní vybavení Přírodovědecké fakulty Univerzity Palackého v Olomouci, a třetí práce je z estonského Tartu, kde student z Institute of Computer Science vizualizoval Delta budovy. Student Marek Příbela [1] svůj model vytvářel téměř celý použitím základních primitiv a byl tedy velmi omezený v možnostech detailů, které lze tímto způsobem vytvořit. Výhodou jeho provedení však je, že má vnitřní části poslucháren, což jsem já z časových důvodů bohužel nestihl, a je to jedna z věcí, které bych rád vytvořil v budoucnu. Model fakulty tohoto studenta nemá žádné okolní budovy ani scenérie. V případě mého modelu také nezbyl čas k vytvoření okolí, proto jsem je nahradil vzdálenými kopci s jehličnany doplněné mlhou pro lepší vizuální požitek. Oproti modelu fakulty

v Olomouci je můj model vytvořený v programu Blender. Student Příbela používal k modelování Unreal Engine.

Na práci Marka Příbely navázal student Tomáš Pospíšil [2], který doplnil model studenta Příbely o drobné předměty, jako jsou například umyvadla, háčky na pověšení kabátů, a podobně. Tento student již pokračoval v Blenderu, a tedy si mohl dovolit použít větší úroveň detailů.

Ze zahraničních prací pro srovnání uvádím vizualizaci Delta budovy, což je budova několika akademických institutů, mezi které patří také Institute of Computer Science. Na této budově předváděl student Aleksander Nikolajev [3] vizualizaci interakce pohybu studentů a učitelů po akademické budově. Ve srovnání s mojí prací se nejednalo o realistickou, ale stylizovanou verzi budovy pro účely předvedení chování osob na fakultě. Zajímavé na této práci je, že údaje o pohybu osob byly snímány senzory pohybu a takto získaná data byla použita v enginu Unity pro vizualizaci chování osob. Student Nikolajev vytvořil stejně jako já vizualizaci budovy v programu Blender a následně ji implementoval v Unity.

5 Závěr

V této bakalářské práci jsem se zabýval různorodými technikami a postupy vizualizace nové budovy Fakulty elektrotechniky a informatiky VŠB-TUO (dále také jako „FEI“) v herním enginu. Úvodní pasáž této práce jsem věnoval základnímu popisu jednotlivých nástrojů aplikace Blender, jež je jednou z 3D modelovacích aplikací. Použití těchto nástrojů jsem demonstroval na ukázce tvorby modelu části budovy fakulty. Dále jsem uvedl problémy, které při modelování vznikly.

Následující pasáž byla věnována problematice PBR materiálů a tvorbě textur. Tuto část jsem doplnil příklady použitých nástrojů z dalšího programu, konkrétně Adobe Photoshop. V rámci závěrečné pasáže jsem se zaměřil na převod konkrétní scény z Blenderu do Unreal Engine 4. I zde jsem práci doplnil o problematické aspekty takového převodu. Kromě toho jsem rovněž na modelu demonstroval možnosti nasvícení scény, nastavení odrazů a odlesků, tvorby terénu, rozmístění vegetace, jakož i nastavení celé interakce se scénou. Kromě toho bylo na ukázce znázorněno plnění určitých úkolů, které souvisejí s obsahem předmětu Základy počítačové grafiky. S ohledem na pandemii COVID-19 nebylo bohužel možné provést testování práce ve virtuální realitě, nicméně technicky je k tomu model připraven a uzpůsoben.

Za nejvýznamnější přínos této práce považuji vytvořený model fakulty, jehož tvorbě jsem se intenzivně věnoval po dobu jednoho roku. I přes veškeré vynaložené úsilí model stále nabízí široké možnosti jeho rozšíření a vylepšení, například přidáním vnitřních částí poslucháren, místností, vytvořením okolních budov kolem fakulty atd. Celý tento projekt může spolu s modelem v Unreal Engine 4 sloužit jako pevný základ pro případné práce dalších studentů, jakož i pro simulace, testování nových funkcí herních engineů, případně může být využit i pro výuku. Považuji za důležité zmínit i skutečnost, že model může být bezpochyby v budoucnu využit k propagaci FEI a jejích aktivit, respektive Vysoké školy báňské – Technické univerzity Ostrava obecně.

Literatura

1. PRÍBELA, Marek. *Budova fakulty v Unreal Engine 4* [online]. Olomouc, 2017 [cit. 2020-05-09]. Dostupné z: <https://stag.upol.cz/portal/studium/prohlizeni.html>. Bakalářská práce. Univerzita Palackého, Fakulta informatiky.
2. POSPÍŠIL, Tomáš. *Interiéry budovy fakulty v Unreal Engine 4* [online]. Olomouc, 2018 [cit. 2020-05-05]. Dostupné z: <https://stag.upol.cz/portal/studium/prohlizeni.html>. Bakalářská práce. Univerzita Palackého v Olomouci, Přírodovědecká fakulta, Katedra informatiky.
3. NIKOLAJEV, Aleksander. *Delta Building Visualisation and Optimisation* [online]. Tartu, Estonsko, 2018 [cit. 2020-05-14]. Dostupné z: https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=61825%5C&year=2018%5C&language=en. Bakalářská práce. UT Institute of Computer Science.
4. *Tip: 6 šikovných aplikací, se kterými můžete bezplatně vytvářet modely pro 3D tisk* [online] [cit. 2020-05-12]. Dostupné z: <https://www.3d-tisk.cz/tip-6-sikovnych-aplikaci-se-kterymi-muzete-bezplatne-vytvaret-modely-pro-3d-tisk/>.
5. *Blender.org* [online]. The Netherlands [cit. 2020-05-04]. Dostupné z: <https://www.blender.org/>.
6. *Blender 2.82 Reference Manual* [online] [cit. 2020-05-14]. Dostupné z: <https://docs.blender.org/manual/en/latest/>.
7. *3D Modeling techniques in Games* [online]. Ukraine: 3D-Ace, 1992-2020 [cit. 2020-05-04]. Dostupné z: <https://3d-ace.com/press-room/articles/3d-modeling-techniques-games>.
8. *Sculpting and Painting* [online] [cit. 2020-05-13]. Dostupné z: https://docs.blender.org/manual/en/latest/sculpt_paint/index.html%5C#modes.
9. MATIÁŠEK, Tomáš. *3D Modelování v Blenderu* [online]. Praha, 2009 [cit. 2020-05-12]. Dostupné z: <https://vskp.vse.cz/eid/14672>. Bakalářská práce. Vysoká škola ekonomická v Praze, Fakulta informatiky a statistiky.
10. *Google Maps (Mapy Google)* [online] [cit. 2020-05-14]. Dostupné z: <https://www.google.cz/maps>.
11. LINDNER, Petr; MYŠKA, Miroslav; TŮMA, Tomáš. *Velká kniha digitální fotografie*. Brno: Computer Press, 2003. ISBN 80-251-0013-8.
12. *The PBR Guide - Part 1* [online] [cit. 2020-05-10]. Dostupné z: <https://academy.substance3d.com/courses/the-pbr-guide-part-1>.
13. *Blueprints Visual Scripting* [online] [cit. 2020-05-08]. Dostupné z: <https://docs.unrealengine.com/en-US/Engine/Blueprints/index.html>.

14. KELBY, Scott. *Digitální fotografie ve Photoshopu: [tipy a techniky používané předními digitálními fotografy]*. Brno: Computer Press, 2003. ISBN 80-722-6990-9.
15. *Přípona souboru* [online] [cit. 2020-05-13]. Dostupné z: <https://soubory.info/extension/psd>.
16. *JPEG* [online] [cit. 2020-05-12]. Dostupné z: <https://jpeg.org/>.
17. *Unrealengine.com* [online]. North Carolina: Epic games, Inc., 2004 [cit. 2020-05-01]. Dostupné z: <https://www.unrealengine.com/>.
18. PV, Sahteesh. *Unreal Engine 4 Game Development Essentials: Master the basics of Unreal Engine 4 to build stunning video games* [online]. 1.vydání. Birmingham, UK: Packt Publishing Ltd., 2016 [cit. 2020-05-04]. ISBN 978-1-78439-196-6. Dostupné z: <https://it-ebooks-search.info/search?q=9781784391966>.
19. *Unity.com* [online] [cit. 2020-05-04]. Dostupné z: <https://unity.com/>.
20. *Cryengine.com* [online] [cit. 2020-05-04]. Dostupné z: <https://www.cryengine.com/>.
21. *Unreal Engine 4 Documentation* [online]. 2004 [cit. 2020-05-14]. Dostupné z: <https://docs.unrealengine.com/en-US/index.html>.
22. MADA, Martin. *Nové možnosti Unreal Engine 4* [online]. Ostrava, 2017 [cit. 2020-05-04]. Dostupné z: https://dspace.vsb.cz/bitstream/handle/10084/119197/MAD0023_FEI_N2647_2612T025_2017.pdf?sequence=1%5C&isAllowed=n. Bakalářská práce. VŠB – Technická univerzita, Ostrava Fakulta elektrotechniky a informatiky, Katedra informatiky.
23. GUŇA, Štěpán. *Možnosti Unreal Engine 4* [online]. Ostrava, 2017 [cit. 2020-05-09]. Dostupné z: <https://dspace.vsb.cz/handle/10084/119194>. Diplomová práce. VŠB – Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky.
24. TRUBAČ, Jan. *Optimalizační techniky v Unreal Engine 4* [online]. Ostrava, 2017 [cit. 2020-05-11]. Dostupné z: <https://dspace.vsb.cz/handle/10084/118974>. Bakalářská práce. VŠB – Technická univerzita Ostrava, Fakulta elektrotechniky a informatiky.

A Seznam příložených obrázků:

- Obrázek č.47: Pohled ze strany poslucháren
- Obrázek č. 48: Pohled ze strany parkoviště
- Obrázek č. 49: Pohled ze strany evakuačního schodiště
- Obrázek č. 50: Pohled na hlavní vchod
- Obrázek č. 51: Pohled na interiér budovy - 2.patro

Výstupem této práce je projekt v Unreal Enginu 4 a několik modelů vytvořených v programu Blender. Kvůli omezené kapacitě pro přílohy k bakalářské práci nebylo možné tento projekt nahrát. Namísto toho byla vytvořena ukázka průchodu fakultou ve formě videa s plněním úkolů a k práci byl přiložen model fakulty v programu Blender. Také byly přiloženy vyjmuté modely a blueprints z projektu v Unreal Enginu 4.



Obrázek 47: Pohled ze strany poslucháren



Obrázek 48: Pohled ze strany parkoviště



Obrázek 49: Pohled ze strany evakuačního schodiště



Obrázek 50: Pohled na hlavní vchod



Obrázek 51: Pohled na interiér budovy - 2.patro